SIAM J. OPTIM.
Vol. 32, No. 3, pp. 1668–1697

POTENTIAL FUNCTION-BASED FRAMEWORK FOR MINIMIZING GRADIENTS IN CONVEX AND MIN-MAX OPTIMIZATION*

JELENA DIAKONIKOLAS† AND PUQIAN WANG†

Abstract. Making the gradients small is a fundamental optimization problem that has eluded unifying and simple convergence arguments in first-order optimization, so far primarily reserved for other convergence criteria, such as reducing the optimality gap. In particular, while many different potential function-based frameworks covering broad classes of algorithms exist for optimality gap-based convergence guarantees, we are not aware of such general frameworks addressing the gradient norm guarantees. To fill this gap, we introduce a novel potential function-based framework to study the convergence of standard methods for making the gradients small in smooth convex optimization and convex-concave min-max optimization. Our framework is intuitive and provides a lens for viewing algorithms that makes the gradients small as being driven by a trade-off between reducing either the gradient norm or a certain notion of an optimality gap. On the lower bounds side, we discuss tightness of the obtained convergence results for the convex setup and provide a new lower bound for minimizing norm of cocoercive operators that allows us to argue about optimality of methods in the min-max setup.

Key words. gradient minimization, convergence analysis, potential function

MSC codes. 90C06, 90C25, 65K05

DOI. 10.1137/21M1395302

1. Introduction.

One of the most basic facts in convex optimization is that a differentiable convex function attains its minimum at a point where its gradient equals zero, provided such a point exists. Thus, it is tempting to conclude that there is no difference between minimizing the function value or its gradient (in any suitable norm). This is only partially true, as we are almost never guaranteed to find a point at which the function is minimized; instead, we opt for a more modest goal of approximating such points. As it turns out, from an algorithmic point of view, there are major differences between guarantees provided for the function value (or optimality gap) and norm of its gradient.

Much of the standard optimization literature on smooth (gradient-Lipschitz) convex first-order optimization has been concerned with providing guarantees for the optimality gap. There is comparatively much less work on guarantees for the norm of the gradient, most of it being initiated after the work of Nesterov [50], which argued that such guarantees are natural and more informative than those based on the function value for certain linearly constrained optimization problems that frequently arise in applications. Further, unlike the optimality gap, which would require knowledge of the minimum function value to be usable as a stopping criterion, the norm of the gradient is readily available to the algorithm as a stopping criterion, as standard first-order methods define their iterates based on the gradient information. This insight is particularly useful for the design of parameter-free algorithms (i.e., algorithms that do

*Received by the editors January 29, 2021; accepted for publication (in revised form) March 11, 2022; published electronically July 21, 2022.
https://doi.org/10.1137/21M1395302
Funding: This research was partially supported by the NSF grant CCF-2007757 and by the Office of the Vice Chancellor for Research and Graduate Education at the University of Wisconsin-Madison, with funding from the Wisconsin Alumni Research Foundation.
†Department of Computer Sciences, University of Wisconsin-Madison, Madison, WI 53706 USA (jelena@cs.wisc.edu, pwang333@wisc.edu).

not require knowledge of function parameters such as smoothness, strong convexity, or sharpness/constants of the Łojasiewicz inequality; see, e.g., [5, 12, 41, 42]), and as such has been used to design parameter-free algorithms that are near-optimal in terms of iteration complexity (i.e., optimal up to poly-logarithmic factors) [29, 40, 51]. The basic idea in these algorithms is to adaptively restart a method with a gradient norm guarantee every time the gradient norm is reduced by constant factor, typically equal to two. The total number of restarts then becomes logarithmic, while the number of iterations between restarts can be bounded using the strong convexity or sharpness assumption, without the need for explicit algorithm knowledge of those constants.

As for L-smooth functions the norm of the gradient can be bounded above as a function of the optimality gap $f(\boldsymbol{x}) - f(\boldsymbol{x}^*)$, where $\boldsymbol{x}^* \in \operatorname{argmin}_{\boldsymbol{x}} f(\boldsymbol{x})$, using

$$\frac{1}{2L}\|\nabla f(\boldsymbol{x})\|^2 \leq f(\boldsymbol{x}) - f(\boldsymbol{x}^*), \tag{1.1}$$

it is not surprising that convergence rates can be established for gradient norm minimization. What *is* surprising, however, is that those rates can be faster than what is implied by (1.1) and existing results for convergence in function value/optimality gap. In particular, methods that are optimal in terms of iteration complexity for minimizing the optimality gap are not necessarily optimal for gradient norm optimization, and vice versa. More specifically, the fast gradient method (FGM) of Nesterov [54] is iteration complexity-optimal for minimizing the optimality gap, but it is suboptimal for minimizing the gradient norm [15, 31].

More generally, the existing literature has not yet shed light on what is the basic mechanism that drives algorithms for gradient norm minimization. The only known iteration complexity-optimal algorithm with respect to (w.r.t.) initial function condition $f(\boldsymbol{x}_0) - f(\boldsymbol{x}^*)$ for minimizing the norm of the gradient of a smooth convex function is due to Kim and Fessler [32].[1] This algorithm was obtained by using the performance estimation (PEP) framework of Drori and Teboulle [22], originally developed for understanding the worst-case performance of optimization algorithms. The algorithm [32] itself and its convergence analysis are inferred from numerical solutions to a semidefinite program (SDP). As such, the intuition behind what is driving the convergence analysis of the algorithm and how the improved convergence rate is obtained is lacking, which constitutes an impediment to possibly generalizing this algorithm to other optimization settings.

Even less is known in the setting of smooth convex-concave min-max optimization, where (near-)optimal convergence results have been established only recently [17, 30, 38, 60, 72] and the problem has been much less studied from the aspect of oracle lower bounds [17, 25, 56]. In particular, similar to the case of convex optimization, classical methods for min-max optimization that are optimal for reducing the primal-dual gap, such as, e.g., the extragradient method [33], mirror-prox [45], and dual extrapolation [49], are suboptimal in terms of iteration complexity for minimizing the gradient norm. Interestingly, however, the methods that turn out to be (near-)optimal were originally studied in the context of fixed point iterations [27, 34, 43].

In this paper, we introduce a novel potential function-based framework to study the convergence in gradient norm for smooth convex and convex-concave optimization problems. Our framework is intuitive, as it relies on establishing convergence of standard methods by interpreting it as a trade-off between reducing the gradient norm and reducing a notion of an optimality gap. The same view can be adopted in a unifying

[1] The optimality of the algorithm can be certified using the lower bounds from, e.g., [15, 47, 48].

manner for methods such as standard gradient descent, Nesterov FGM [54], the optimized method of Kim and Fessler [32], gradient descent-ascent (which is equivalent to the Krasnosel'skiı–Mann iteration [34, 43]; see section 3.1), and Halpern iteration [27]. We further complement these results with a discussion of optimality of the considered methods for convex optimization, and with a new lower bound for minimizing the norm of cocoercive operators (see section 1.2 for a precise definition and relationship to min-max optimization), which allows us to discuss optimality of gradient descent-ascent and Halpern iteration as methods for minimizing the gradient norm in smooth convex-concave min-max optimization.

1.1. Further related work. Understanding the phenomenon of acceleration and providing a unifying theory of first-order optimization algorithms, often based on the use of potential functions, has been an important topic in optimization research, with a flurry of recent research activity in this area [1, 4, 6, 7, 8, 9, 11, 14, 19, 20, 21, 24, 28, 35, 37, 39, 61, 62, 63, 65, 66, 69, 70, 71, 74]. However, the existing literature has almost exclusively focused on the optimality gap guarantees, with only a small subset of results seeking to provide guarantees for gradient norm and primarily addressing FGM-type algorithms with suboptimal rates [6, 19, 44, 62]. An exception is [36], which appeared subsequent to our work. In particular, [36] provided a geometric interpretation of acceleration, which allowed the authors to construct potential functions that led to a method that generalizes [32] to the setting of composite (smooth plus nonsmooth) optimization.

Complementary to the literature discussed above, whose focus has been on deriving intuitive convergence analysis frameworks, another line of work has focused on using the SDP-based performance estimation framework of Drori and Teboulle [22] to investigate the worst-case performance of optimization algorithms [16, 30, 31, 32, 38, 67, 68]. Most relevant to our work among these results are the following: [31], which investigated the worst-case performance of FGM-type methods in terms of gradient norm minimization; [32], which obtained the first (and so far, the only) iteration complexity-optimal algorithm for minimizing the gradient norm of smooth convex functions; and [38], which obtained a tight worst-case convergence bound for Halpern iteration. While the SDP-based approach used in this line of work is useful for understanding the worst-case performance of existing algorithms (and even obtaining new algorithms [32]), its downside is that, because the convergence arguments are computer-assisted (namely, they are inferred from numerical solutions to SDPs), they are generally not suitable for developing intuition about what is driving the methods and their analysis.

Finally, from the aspect of lower bounds, gradient norm minimization is well understood in the setting of convex optimization [15, 47, 48]. For min-max optimization, [17] provided near-tight lower bounds in the first-order oracle model, using the lower bound for optimality gap from [56] and algorithmic reductions between different problem classes. The lower bounds provided in this paper are tight; however, they apply to the more restricted complexity model, where the algorithm iterates are required to lie in the span of previously queried gradients [47, 48]. To obtain these lower bounds, we build on the techniques developed in [2, 3]. Subsequent to our work, [72] also provided similar tight lower bounds in the same complexity model by establishing a connection between the lower bounds for solving linear systems of equations from [47, 48] and biaffine min-max optimization problems. The work [72] has also discussed how to generalize such lower bounds to the gradient oracle model.

1.2. Notation and preliminaries. Throughout this paper, we consider the Euclidean space $(\mathbb{R}^d, \|\cdot\|)$, where $\|\cdot\| = \sqrt{\langle\cdot,\cdot\rangle}$ is the Euclidean norm and $\langle\cdot,\cdot\rangle$ denotes any inner product on $\mathbb{R}^d$. We use $\{A_k\}_{k\geq 0}$ and $\{B_k\}_{\geq 0}$ to denote sequences of nondecreasing, nonnegative numbers, and define $a_0 = A_0$, $a_k = A_k - A_{k-1}$ for $k \geq 1$, and, similarly, $b_0 = B_0$, $b_k = B_k - B_{k-1}$ for $k \geq 1$.

We consider two main problem setups: (i) making the gradients small in convex optimization, and (ii) making the gradients small in min-max optimization.

Convex optimization. In the first setup, we assume we are given first-order oracle access to a convex continuously differentiable function $f : \mathbb{R}^d \to \mathbb{R}$. The first-order definition of convexity then applies, and we have

$$(\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d): \quad f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle\,.$$

We further assume that f is L-smooth, i.e., that its gradients are L-Lipschitz continuous:

$$(\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d): \quad \|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|.$$

Recall that smoothness of f implies

$$(\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d): \quad f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle + \frac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2. \tag{1.2}$$

The goal of the first setup is to, given $\epsilon > 0$, construct a point $\boldsymbol{x}$ such that $\|\nabla f(\boldsymbol{x})\| \leq \epsilon$ in as few iterations (oracle queries to the gradient of f) as possible. A useful fact that turns out to be crucial for the analysis in the convex case is the following (see, e.g., [73, section 3.5]).

FACT 1.1. *A continuously differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is L-smooth and convex if and only if*

$$(\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d): \quad \frac{1}{2L}\|\nabla f(\boldsymbol{y}) - \nabla f(\boldsymbol{x})\|^2 \leq f(\boldsymbol{y}) - f(\boldsymbol{x}) - \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle\,. \tag{1.3}$$

Observe that Fact 1.1 fully characterizes the class of smooth convex functions, and, as such, should be sufficient for analyzing any algorithm that addresses problems from this class.[2] An immediate consequence of Fact 1.1 is that the gradient of a smooth convex function is cocoercive, i.e.,

$$\langle \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y}\rangle \geq \frac{1}{L}\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|^2. \tag{1.4}$$

Min-max optimization. In the second setup, we are given oracle access to gradients of a function $\phi : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{R}$, where $d_1 + d_2 = d$. Function $\phi(\boldsymbol{x}, \boldsymbol{y})$ is assumed to be convex-concave: convex in the first argument ($\boldsymbol{x}$) when the second argument ($\boldsymbol{y}$) is fixed and concave in the second argument ($\boldsymbol{y}$) when the first argument ($\boldsymbol{x}$) is fixed, for any values of $\boldsymbol{x}, \boldsymbol{y}$. Similar to the case of convex optimization, the goal in this case is, given $\epsilon > 0$, to find a pair of points $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$ such that $\|\nabla\phi(\boldsymbol{x}, \boldsymbol{y})\| \leq \epsilon$ in as few iterations (oracle queries to the gradient of ϕ) as possible.

[2] This statement can be further formalized to claim that, in fact, the inequality from (1.3) evaluated only at the iterates and at the optimum is sufficient for the analysis of almost all standard algorithms for smooth convex optimization. This is a consequence of the tightness of PEPs from [68].

We consider the problem of minimizing the norm of the gradient of ϕ as the problem of minimizing the norm of the operator $F(\boldsymbol{u}) = \left[\begin{smallmatrix} \nabla_{\boldsymbol{x}}\phi(\boldsymbol{x},\boldsymbol{y}) \\ -\nabla_{\boldsymbol{y}}\phi(\boldsymbol{x},\boldsymbol{y}) \end{smallmatrix}\right]$, where $\boldsymbol{u} = \left[\begin{smallmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{smallmatrix}\right]$. When ϕ is convex-concave, F is monotone [58], i.e., it holds that

$$(\forall \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d): \quad \langle F(\boldsymbol{u}) - F(\boldsymbol{v}), \boldsymbol{u} - \boldsymbol{v} \rangle \geq 0. \tag{1.5}$$

We will assume throughout that F is $\frac{1}{L}$-cocoercive, i.e., that

$$(\forall \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d): \quad \langle F(\boldsymbol{u}) - F(\boldsymbol{v}), \boldsymbol{u} - \boldsymbol{v} \rangle \geq \frac{1}{L}\|F(\boldsymbol{u}) - F(\boldsymbol{v})\|^2. \tag{1.6}$$

Cocoercivity of F implies that it is monotone and L-Lipschitz. The opposite does not hold, in general, unless F is the gradient of a smooth convex function (as we saw in the case of convex optimization described earlier). Nevertheless, cocoercivity is sufficient to capture the main algorithmic ideas of smooth min-max optimization, and the extensions to general smooth min-max optimization are possible through the use of approximate resolvent operators (see, e.g., [17]). Further, it suffices to consider unconstrained problems, as extensions to constrained optimization problems are possible in a straightforward manner using a notion of *operator mapping* (see, e.g., [17], where a similar idea was used).

We assume here that there exists a point $\boldsymbol{u}^* \in \mathbb{R}^d$ such that $F(\boldsymbol{u}^*) = 0$. Due to cocoercivity of F (see (1.6)), this assumption implies that

$$(\forall \boldsymbol{u} \in \mathbb{R}^d): \quad \langle F(\boldsymbol{u}), \boldsymbol{u} - \boldsymbol{u}^* \rangle \geq \frac{1}{L}\|F(\boldsymbol{u})\|^2. \tag{1.7}$$

It will be useful to think of $\langle F(\boldsymbol{u}), \boldsymbol{u} - \boldsymbol{u}^* \rangle$ as a notion of "optimality gap" for min-max optimization problems, as, using convexity-concavity of ϕ, we have

$$(\forall (\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2}): \quad \phi(\boldsymbol{x}, \boldsymbol{y}^*) - \phi(\boldsymbol{x}, \boldsymbol{y}) + \phi(\boldsymbol{x}, \boldsymbol{y}) - \phi(\boldsymbol{x}^*, \boldsymbol{y}) \leq \langle F(\boldsymbol{u}), \boldsymbol{u} - \boldsymbol{u}^* \rangle .$$

2. Small gradients in convex optimization. In this section, we consider the problem of minimizing the norm of the gradient of a smooth convex function. We show that all standard methods, including standard gradient descent, the fast gradient method of Nesterov [54], and the optimized gradient method of Kim and Fessler [32], can be captured within an intuitive potential function-based framework, where the progress of a method is established through a trade-off between the norm of the gradient and the optimality gap. Further, the complete convergence analysis of each of the methods can be fully carried out using only the cocoercivity inequality from (1.3), which fully characterizes the class of smooth convex functions.

2.1. Gradient descent. As a warmup, we start by considering L-smooth but possibly nonconvex objectives f. In this case, all that can be said about f is that its gradients are L-Lipschitz, which implies (1.2). Further, for any descent-type method, we cannot hope to bound the norm of the last gradient—all that we can hope for is the average or the minimum over all seen gradients. The simplest way to see this is by considering the one dimensional case: if the function is locally concave and the algorithm moves in the direction that reduces the function value, the absolute value of the function derivative must increase.

Thus, assuming that the function is bounded below by some $f_\star > -\infty$, it is natural to consider methods that in each iteration either reduce the function value or the norm of the gradient. Such methods ensure that $\forall k \geq 0$,

$$a_k\|\nabla f(\boldsymbol{x}_k)\|^2 + f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k) \leq 0,$$

or, equivalently, that the potential function

$$\mathcal{C}_k = \sum_{i=0}^{k} a_i \|\nabla f(\boldsymbol{x}_i)\|^2 + f(\boldsymbol{x}_{k+1}) \tag{2.1}$$

is nonincreasing, where a_i is some sequence of positive numbers. Equivalently, such methods ensure that $a_k\|\nabla f(\boldsymbol{x}_k)\|^2 + f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k) \leq 0\ \forall k \geq 0$.

As the only assumption we are making about f is that it is L-smooth, the most we can do to bound $f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)$ is use (1.2). The tightest bound on $f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)$ that can be obtained from (1.2) is attained when $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)$ (i.e., for the standard gradient descent step) and is given by $f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k) \leq -\frac{1}{2L}\|\nabla f(\boldsymbol{x}_k)\|^2$, in which case the largest a_k we can choose is $a_k = \frac{1}{2L}$.[3] As $\mathcal{C}_k$ is nonincreasing, it follows that $\mathcal{C}_k \leq \mathcal{C}_0$, and we recover the familiar convergence bound of gradient descent:

$$\frac{1}{k+1}\sum_{i=0}^{k} \|\nabla f(\boldsymbol{x}_i)\|^2 \leq \frac{2L(f(\boldsymbol{x}_0) - f(\boldsymbol{x}_{k+1}))}{k+1} \leq \frac{2L(f(\boldsymbol{x}_0) - f_\star)}{k+1}. \tag{2.2}$$

When considering the case of a convex objective function f, the first question to ask is how would convexity help to improve the bound from (2.2). The first observation to make is that Fact 1.1 fully characterizes the class of smooth convex functions, and, thus, (1.3) should be enough to carry out the analysis of any algorithm for smooth convex functions.

Given that the function is convex, in this case it seems reasonable to hope that we can obtain a bound on the gradient norm at the last iterate. Thus, we could consider a potential function of the form

$$\mathcal{C}_k = A_k\|\nabla f(\boldsymbol{x}_k)\|^2 + f(\boldsymbol{x}_k)$$

and try enforcing the condition that $\mathcal{C}_k \leq \mathcal{C}_{k-1}$ for A_k that grows as fast as possible with the iteration count k. This approach precisely gives the bound $\|\nabla f(\boldsymbol{x}_k)\|^2 \leq \frac{2L(f(\boldsymbol{x}_0) - f(\boldsymbol{x}^*))}{2k+1}$, which is tight (see, e.g., [32, Lemma 5.2]). While such an upper bound was already proved in [32, Theorem 5.1] using the PEP framework [22], we note that the argument we provide here is directly motivated by the trade-off between minimizing the gradient norm and the optimality gap, which is the main unifying feature of the potential functions provided in our work.

LEMMA 2.1 (convergence of gradient descent). *Let $f : \mathbb{R}^d \to \mathbb{R}$ be an L-smooth function that attains its minimum on $\mathbb{R}^d$, and let $\boldsymbol{x}^* \in \operatorname{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$. Let $\boldsymbol{x}_0 \in \mathbb{R}^d$ be an arbitrary initial point and assume that the sequence $\{\boldsymbol{x}_k\}_{k\geq 0}$ evolves according to the standard gradient descent, i.e., $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)\ \forall k \geq 0$. Then*

$$\mathcal{C}_k = \frac{k}{L}\|\nabla f(\boldsymbol{x}_k)\|^2 + f(\boldsymbol{x}_k)$$

is nonincreasing with k, and we can conclude that $\forall k \geq 0$

$$\|\nabla f(\boldsymbol{x}_k)\|^2 \leq \frac{2L(f(\boldsymbol{x}_0) - f(\boldsymbol{x}^*))}{2k+1}.$$

[3] This specific inequality is also known as the "descent lemma." Analysis of gradient descent based on telescoping this inequality appears in standard optimization texts.

Proof. We start by showing that $\mathcal{C}_{k+1} \leq \mathcal{C}_k \ \forall k \geq 0$. By the definition of $\mathcal{C}_k$,

$$\mathcal{C}_{k+1} - \mathcal{C}_k \leq \frac{k+1}{L}\|\nabla f(\boldsymbol{x}_{k+1})\|^2 - \frac{k}{L}\|\nabla f(\boldsymbol{x}_k)\|^2 + f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k).$$

Applying Fact 1.1 with $\boldsymbol{x} = \boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)$ and $\boldsymbol{y} = \boldsymbol{x}_k$, it follows that $f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k) \leq -\frac{1}{2L}\|\nabla f(\boldsymbol{x}_{k+1})\|^2 - \frac{1}{2L}\|\nabla f(\boldsymbol{x}_k)\|^2$, and thus

$$\mathcal{C}_{k+1} - \mathcal{C}_k \leq \frac{2k+1}{2L}\|\nabla f(\boldsymbol{x}_{k+1})\|^2 - \frac{2k+1}{2L}\|\nabla f(\boldsymbol{x}_k)\|^2.$$

To complete the proof that $\mathcal{C}_{k+1} \leq \mathcal{C}_k$, it remains to argue that $\|\nabla f(\boldsymbol{x}_{k+1})\| \leq \|\nabla f(\boldsymbol{x}_k)\| \ \forall k \geq 0$. This is clearly true if $\|\nabla f(\boldsymbol{x}_{k+1})\| = 0$, so assume $\|\nabla f(\boldsymbol{x}_{k+1})\| \neq 0$. Applying (1.4) with $\boldsymbol{x} = \boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)$, $\boldsymbol{y} = \boldsymbol{x}_k$, and simplifying, it follows that

$$\|\nabla f(\boldsymbol{x}_{k+1})\|^2 \leq \langle \nabla f(\boldsymbol{x}_{k+1}), \nabla f(\boldsymbol{x}_k)\rangle \leq \|\nabla f(\boldsymbol{x}_{k+1})\|\|\nabla f(\boldsymbol{x}_k)\|,$$

where the last inequality is by Cauchy–Schwarz. To conclude that $\|\nabla f(\boldsymbol{x}_{k+1})\| \leq \|\nabla f(\boldsymbol{x}_k)\|$, it remains to divide both sides of the last inequality by $\|\nabla f(\boldsymbol{x}_{k+1})\|$.

From the first part of the proof, it follows that $\mathcal{C}_k \leq \mathcal{C}_0$, and thus

$$\frac{k}{L}\|\nabla f(\boldsymbol{x}_k)\|^2 \leq f(\boldsymbol{x}_0) - f(\boldsymbol{x}_k) = f(\boldsymbol{x}_0) - f(\boldsymbol{x}^*) + f(\boldsymbol{x}^*) - f(\boldsymbol{x}_k).$$

It remains to observe that $f(\boldsymbol{x}^*) - f(\boldsymbol{x}_k) \leq -\frac{1}{2L}\|\nabla f(\boldsymbol{x}_k)\|^2$, which follows by applying Fact 1.1 with $\boldsymbol{x} = \boldsymbol{x}^*$, $\boldsymbol{y} = \boldsymbol{x}_k$, and rearranging. ∎

2.2. Methods that are faster than gradient descent. The potential functions we have seen so far (for gradient descent) trade off the gradient norm (squared) with the function value. Equivalently, we can view them as trading off the gradient norm with the optimality gap $f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*)$, as $f(\boldsymbol{x}^*)$ would cancel out in the analysis and the same argument would go through.

It is reasonable to ask whether we can obtain faster algorithms by using a different trade-off, say, by considering potential functions of the form $\mathcal{C}_k = A_k\|\nabla f(\boldsymbol{x}_k)\|^2 + B_k(f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*))$ or $\mathcal{C}_k = \sum_{i=0}^{k} a_i\|\nabla f(\boldsymbol{x}_i)\|^2 + B_k(f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*))$, where B_k is some positive function of the iteration count k.

Observe that for nonconstant B_k, one way or another, we would need to account for $\boldsymbol{x}^*$, which is not known to the algorithm. This happens because in $\mathcal{C}_k - \mathcal{C}_{k-1}$ with nonconstant B_k, $f(\boldsymbol{x}^*)$ does not get cancelled out. However, there are at least two ways around this issue. The first one is to utilize (1.3) to bound below $f(\boldsymbol{x}^*)$. This approach does not lead to the optimal iteration complexity, but improves the overall bound compared to gradient descent and recovers a variant of Nesterov FGM. The second approach is to replace the optimality gap with a gap to some reference point. In particular, as we show below, optimized gradient method [32] can be viewed as using the final point of the algorithm $\boldsymbol{x}_N$ as the reference (or anchor) point.

2.2.1. Fast gradient method. We start by considering a potential function that offers a different trade-off between the norm of the gradient and the optimality gap, defined by

$$\mathcal{C}_k = \sum_{i=0}^{k-1} a_i\|\nabla f(\boldsymbol{x}_i)\|^2 + B_k(f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*)), \tag{2.3}$$

where $a_i > 0\ \forall i \geq 0$ and the sequence of scalars $B_k > 0\ \forall k \geq 0$ is strictly increasing. We also define $b_k = B_k - B_{k-1} > 0$. By convention, the summation from i to j where $j < i$ is taken to be zero. Observe that

$$\mathcal{C}_0 = B_0(f(\boldsymbol{x}_0) - f(\boldsymbol{x}^*)). \tag{2.4}$$

While, in principle, one could also consider $\mathcal{C}_k = A_k\|\nabla f(\boldsymbol{x}_k)\|^2 + B_k(f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*))$ hoping to obtain a bound on the last gradient, it is not clear that such a bound is even possible for nonconstant B_k (see section 2.3).

We first show that there is a natural algorithm that ensures $\mathcal{C}_{k+1} - \mathcal{C}_k \leq E_k\ \forall k \geq 0$, where E_k contains only telescoping terms. As it turns out, this algorithm is precisely Nesterov FGM.

LEMMA 2.2. *Given an arbitrary initial point $\boldsymbol{x}_0 \in \mathbb{R}^d$, assume that for $k \geq 1$, the sequence $\boldsymbol{x}_k$ is updated as*

$$\boldsymbol{x}_k = \frac{B_{k-1}}{B_k}\Big(\boldsymbol{x}_{k-1} - \frac{1}{L}\nabla f(\boldsymbol{x}_{k-1})\Big) + \frac{b_k}{B_k}\boldsymbol{v}_k, \tag{2.5}$$

where $\boldsymbol{v}_k$ is defined recursively via $\boldsymbol{v}_k = \boldsymbol{v}_{k-1} - \frac{b_{k-1}}{L}\nabla f(\boldsymbol{x}_{k-1})$ with $\boldsymbol{v}_0 = \boldsymbol{x}_0$. If ${b_k}^2 \leq B_k$ and $a_{k-1} \leq \frac{B_{k-1}}{2L}$, then $\mathcal{C}_k - \mathcal{C}_{k-1} \leq \frac{L}{2}\big(\|\boldsymbol{x}^* - \boldsymbol{v}_k\|^2 - \|\boldsymbol{x}^* - \boldsymbol{v}_{k+1}\|^2\big)\ \forall k \geq 1$, where $\mathcal{C}_k$ is defined by (2.3).

Proof. Given $k \geq 1$, by definition of $\mathcal{C}_k$, we have

$$\mathcal{C}_k - \mathcal{C}_{k-1} = a_{k-1}\|\nabla f(\boldsymbol{x}_{k-1})\|^2 + B_k f(\boldsymbol{x}_k) - B_{k-1}f(\boldsymbol{x}_{k-1}) - b_k f(\boldsymbol{x}^*). \tag{2.6}$$

Since $f(\boldsymbol{x}^*)$ is not known to the algorithm and we are trying to bound $\mathcal{C}_k - \mathcal{C}_{k-1}$ above, it appears natural to use (1.3) to bound $f(\boldsymbol{x}^*)$ below. In particular, we have

$$f(\boldsymbol{x}^*) \geq f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x}^* - \boldsymbol{x}_k\rangle + \frac{1}{2L}\|\nabla f(\boldsymbol{x}_k)\|^2. \tag{2.7}$$

On the other hand, the difference $f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k-1})$ can be bounded above using, again, (1.3), as follows:

$$\begin{aligned} f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k-1}) \leq\ & \left\langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x}_k - \boldsymbol{x}_{k-1} + \frac{1}{L}\nabla f(\boldsymbol{x}_{k-1})\right\rangle \\ & - \frac{1}{2L}\|\nabla f(\boldsymbol{x}_k)\|^2 - \frac{1}{2L}\|\nabla f(\boldsymbol{x}_{k-1})\|^2. \end{aligned} \tag{2.8}$$

Combining (2.7) and (2.8) with (2.6), we have

$$\begin{aligned} \mathcal{C}_k - \mathcal{C}_{k-1} \leq\ & -\frac{B_k}{2L}\|\nabla f(\boldsymbol{x}_k)\|^2 + \Big(a_{k-1} - \frac{B_{k-1}}{2L}\Big)\|\nabla f(\boldsymbol{x}_{k-1})\|^2 \\ & + B_{k-1}\left\langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x}_k - \boldsymbol{x}_{k-1} + \frac{1}{L}\nabla f(\boldsymbol{x}_{k-1})\right\rangle + b_k\left\langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x}_k - \boldsymbol{x}^*\right\rangle. \end{aligned} \tag{2.9}$$

Now, if b_k were zero (constant B_k), we could simply set $\boldsymbol{x}_k = \boldsymbol{x}_{k-1} - \frac{1}{L}\nabla f(\boldsymbol{x}_{k-1})$, and we would be recovering gradient descent and its analysis from the previous subsection. Of course, the goal here is to get a different trade-off, where B_k is strictly increasing.

To get a useful bound on $\mathcal{C}_k - \mathcal{C}_{k-1}$, we need to be able to bound or otherwise control the term $b_k\left\langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x}_k - \boldsymbol{x}^*\right\rangle$. Fortunately, such a term frequently appears

in the mirror-descent-type analysis, and it can be bounded using standard arguments by defining

$$\begin{aligned}\boldsymbol{v}_{k+1} &= \operatorname*{argmin}_{\boldsymbol{u}\in\mathbb{R}^d}\left\{b_k\left\langle\nabla f(\boldsymbol{x}_k),\boldsymbol{u}-\boldsymbol{v}_k\right\rangle+\frac{L}{2}\|\boldsymbol{u}-\boldsymbol{v}_k\|^2\right\}\\ &= \boldsymbol{v}_k-\frac{b_k}{L}\nabla f(\boldsymbol{x}_k).\end{aligned}$$

Then, we have

$$\begin{aligned}b_k\left\langle\nabla f(\boldsymbol{x}_k),\boldsymbol{x}_k-\boldsymbol{x}^*\right\rangle &= b_k\left\langle\nabla f(\boldsymbol{x}_k),\boldsymbol{x}_k-\boldsymbol{v}_{k+1}\right\rangle+L\left\langle\boldsymbol{v}_k-\boldsymbol{v}_{k+1},\boldsymbol{v}_{k+1}-\boldsymbol{x}^*\right\rangle\\ &= b_k\left\langle\nabla f(\boldsymbol{x}_k),\boldsymbol{x}_k-\boldsymbol{v}_k\right\rangle+\frac{{b_k}^2}{L}\|\nabla f(\boldsymbol{x}_k)\|^2\\ &\quad+\frac{L}{2}\|\boldsymbol{x}^*-\boldsymbol{v}_k\|^2-\frac{L}{2}\|\boldsymbol{x}^*-\boldsymbol{v}_{k+1}\|^2-\frac{L}{2}\|\boldsymbol{v}_{k+1}-\boldsymbol{v}_k\|^2\\ &= b_k\left\langle\nabla f(\boldsymbol{x}_k),\boldsymbol{x}_k-\boldsymbol{v}_k\right\rangle+\frac{{b_k}^2}{2L}\|\nabla f(\boldsymbol{x}_k)\|^2\\ &\quad+\frac{L}{2}\|\boldsymbol{x}^*-\boldsymbol{v}_k\|^2-\frac{L}{2}\|\boldsymbol{x}^*-\boldsymbol{v}_{k+1}\|^2,\end{aligned}$$

where we have repeatedly used $\boldsymbol{v}_{k+1}=\boldsymbol{v}_k-\frac{b_k}{L}\nabla f(\boldsymbol{x}_k)$. Combining with (2.9), we have

$$\begin{aligned}\mathcal{C}_k-\mathcal{C}_{k-1} &\le \frac{{b_k}^2-B_k}{2L}\|\nabla f(\boldsymbol{x}_k)\|^2+\Big(a_{k-1}-\frac{B_{k-1}}{2L}\Big)\|\nabla f(\boldsymbol{x}_{k-1})\|^2\\ &\quad+\frac{L}{2}\|\boldsymbol{x}^*-\boldsymbol{v}_k\|^2-\frac{L}{2}\|\boldsymbol{x}^*-\boldsymbol{v}_{k+1}\|^2\\ &\quad+\left\langle\nabla f(\boldsymbol{x}_k),B_k\boldsymbol{x}_k-B_{k-1}\Big(\boldsymbol{x}_{k-1}-\frac{1}{L}\nabla f(\boldsymbol{x}_{k-1})\Big)-b_k\boldsymbol{v}_k\right\rangle.\end{aligned}$$

To obtain $\mathcal{C}_k-\mathcal{C}_{k-1}\le\frac{L}{2}\|\boldsymbol{x}^*-\boldsymbol{v}_k\|^2-\frac{L}{2}\|\boldsymbol{x}^*-\boldsymbol{v}_{k+1}\|^2$, it remains to choose ${b_k}^2\le B_k$, $a_{k-1}\le\frac{B_{k-1}}{2L}$, and $\boldsymbol{x}_k=\frac{B_{k-1}}{B_k}\big(\boldsymbol{x}_{k-1}-\frac{1}{L}\nabla f(\boldsymbol{x}_{k-1})\big)+\frac{b_k}{B_k}\boldsymbol{v}_k$. ∎

We can now use Lemma 2.2 to argue about convergence of Nesterov FGM from (2.5). Interestingly, the result from Lemma 2.2 suffices to argue about both convergence in function value and in norm of the gradient. Although both bounds are known (see, e.g., [19, 31, 54, 62]), to the best of our knowledge, this is the first analysis that simultaneously leads to both convergence guarantees.

THEOREM 2.3 (convergence of fast gradient method). *Suppose that the assumptions of Lemma 2.2 hold, where $\boldsymbol{v}_0=\boldsymbol{x}_0$. Then $\forall k\ge 1$,*

$$f(\boldsymbol{x}_k)-f(\boldsymbol{x}^*)\le\frac{2B_0(f(\boldsymbol{x}_0)-f(\boldsymbol{x}^*))+L\|\boldsymbol{x}_0-\boldsymbol{x}^*\|^2}{2B_k}$$

and

$$\sum_{i=0}^{k}a_i\|\nabla f(\boldsymbol{x}_i)\|^2\le B_0(f(\boldsymbol{x}_0)-f(\boldsymbol{x}^*))+\frac{L}{2}\|\boldsymbol{x}_0-\boldsymbol{x}^*\|^2.$$

In particular, if $b_0=B_0=1$, ${b_k}^2=B_k$ for $k\ge1$, and $a_k=\frac{B_k}{2L}$, then

$$f(\boldsymbol{x}_k)-f(\boldsymbol{x}^*)\le\frac{4L\|\boldsymbol{x}_0-\boldsymbol{x}^*\|^2}{(k+1)(k+2)}$$

and

$$\min_{0\le i\le k}\|\nabla f(\boldsymbol{x}_i)\|^2 \le \frac{\sum_{i=0}^k B_i\|\nabla f(\boldsymbol{x}_i)\|^2}{\sum_{i=0}^k B_i} \le \frac{18L^2\|\boldsymbol{x}_0-\boldsymbol{x}^*\|^2}{(k+1)(k+2)(k+3)}.$$

Proof. Applying Lemma 2.2 and the definition of $\mathcal{C}_k$, we have $\forall k \ge 1$,

$$\begin{aligned}\mathcal{C}_k &\le \mathcal{C}_0 + \frac{L}{2}\|\boldsymbol{x}^*-\boldsymbol{v}_0\|^2 - \frac{L}{2}\|\boldsymbol{v}_{k+1}-\boldsymbol{x}^*\|^2\\ &\le B_0(f(\boldsymbol{x}_0)-f(\boldsymbol{x}^*)) + \frac{L}{2}\|\boldsymbol{x}^*-\boldsymbol{x}_0\|^2.\end{aligned}$$

Equivalently,

$$\sum_{i=0}^{k-1} a_i\|\nabla f(\boldsymbol{x}_i)\|^2 + B_k(f(\boldsymbol{x}_k)-f(\boldsymbol{x}^*)) \le B_0(f(\boldsymbol{x}_0)-f(\boldsymbol{x}^*)) + \frac{L}{2}\|\boldsymbol{x}^*-\boldsymbol{x}_0\|^2.$$

The first part of the theorem is now immediate, as $\sum_{i=0}^{k-1} a_i\|\nabla f(\boldsymbol{x}_i)\|^2 \ge 0$ and

$$B_k(f(\boldsymbol{x}_k)-f(\boldsymbol{x}^*)) \ge \frac{B_k}{2L}\|\nabla f(\boldsymbol{x}_k)\|^2 \ge a_k\|\nabla f(\boldsymbol{x}_k)\|^2.$$

For the second part, we need only bound the growth of B_k when ${b_k}^2 = (B_k - B_{k-1})^2 = B_k$. It is a standard result that this growth is quadratic and at least as fast as the growth resulting from choosing $b_k = \frac{k+1}{2}\ \forall k$. Thus, $B_k \ge \sum_{i=0}^k \frac{i+1}{2} = \frac{(k+1)(k+2)}{4}$ and $\sum_{i=0}^k B_i \ge \frac{(k+1)(k+2)(k+3)}{12}$. Using that $f(\boldsymbol{x}_0)-f(\boldsymbol{x}^*) \le \frac{L}{2}\|\boldsymbol{x}_0-\boldsymbol{x}^*\|^2$, it now follows from the first part of the theorem that

$$f(\boldsymbol{x}_k)-f(\boldsymbol{x}^*) \le \frac{4L\|\boldsymbol{x}_0-\boldsymbol{x}^*\|^2}{(k+1)(k+2)}$$

and

$$\min_{0\le i\le k}\|\nabla f(\boldsymbol{x}_i)\|^2 \le \frac{\sum_{i=0}^k B_i\|\nabla f(\boldsymbol{x}_i)\|^2}{\sum_{i=0}^k B_i} \le \frac{18L^2\|\boldsymbol{x}_0-\boldsymbol{x}^*\|^2}{(k+1)(k+2)(k+3)},$$

as claimed. ∎

The bounds presented in Theorem 2.3 are tight in the worst-case sense, up to small multiplicative constants. The tightness of the $O(1/k^2)$ convergence bound for the optimality gap is well known [46, 52], with a recent lower bound of Drori [23] even providing a tight constant. The tightness of the $O(1/k^3)$ bound for the minimum squared norm of the gradient was demonstrated numerically in [31, Table 3] and [68, section 4.3].

Remark 2.4. It may not be immediately clear why the bound from Theorem 2.3 improves upon the bound for gradient descent from Lemma 2.1, as in the former the gradient is bounded as a function of $\|\boldsymbol{x}^*-\boldsymbol{x}_0\|^2$, while in the latter it is bounded as a function of $f(\boldsymbol{x}_0)-f(\boldsymbol{x}^*)$. Here, one should note that, using the standard convergence result for the optimality gap of gradient descent $f(\boldsymbol{x}_k)-f(\boldsymbol{x}^*) = O\big(\frac{L\|\boldsymbol{x}_0-\boldsymbol{x}^*\|^2}{k}\big)$ and combining it with the bound from Lemma 2.1, we also have that $\|\nabla f(\boldsymbol{x}_k)\|^2 = O\big(L(\frac{f(\boldsymbol{x}_{\lceil k/2\rceil})-f(\boldsymbol{x}^*)}{k})\big) = O\big(\frac{L^2\|\boldsymbol{x}_0-\boldsymbol{x}^*\|^2}{k^2}\big)$.[4] Furthermore, this bound is known to be

[4]Note that such an upper bound can also be obtained directly, using the potential function-based argument from [66] or as a special case of the result for GDA presented in section 3.1.

tight [31, Theorem 2], and it also applies to $\min_{0\le i\le k}\|\nabla f(\boldsymbol{x}_i)\|^2$, as gradient descent monotonically decreases the gradient. We also note that the improved bound for FGM from Theorem 2.3 can only be established for the minimum gradient norm up to iteration k; as shown numerically in [31, Table 4] and [68, Table 4], the bound for the gradient of the last iterate is no better than that of gradient descent, i.e., $\|\nabla f(\boldsymbol{x}_k)\|^2 = \Omega\big(\frac{L^2\|\boldsymbol{x}_0-\boldsymbol{x}^*\|^2}{k^2}\big)$.

2.2.2. Optimized method for the gradients. The only known method that achieves the optimal convergence bound of the form $\|\nabla f(\boldsymbol{x}_k)\|^2 = O\big(\frac{L(f(\boldsymbol{x}_0)-f(\boldsymbol{x}^*))}{k^2}\big)$ is the optimized method for the gradients (OGM-G), due to Kim and Fessler [32]. This method was obtained using the PEP framework of Drori and Teboulle [22], which relies on numerical solutions to semidefinite programs that model the worst-case performance of methods on a given class of problems (such as, e.g., unconstrained problems with smooth convex objective functions considered here). While this is a very powerful approach that generally produces tight convergence analysis and worst-case instances as a byproduct, as discussed before the intuition behind the methods and their analysis obtained using PEP is not always clear.

In this section, we show that OGM-G is a direct consequence arising from a potential function that fits within the broader framework studied in this paper. In particular, as mentioned earlier in this section, we can view OGM-G as trading off the norm of the gradient for a gap w.r.t. an anchor point, which is the last point constructed by the algorithm. As a consequence of anchoring to the last point, the algorithm crucially requires fixing the number of iterations in advance to achieve the optimal convergence bound stated above.

The potential function used for analyzing OGM-G is defined by

$$\mathcal{C}_k = A_k\Big(\frac{1}{2L}\|\nabla f(\boldsymbol{x}_k)\|^2 + \frac{1}{2L}\|\nabla f(\boldsymbol{x}_K)\|^2 + f(\boldsymbol{x}_k) - f(\boldsymbol{x}_K)\Big), \tag{2.10}$$

where K is the total number of iterations for which OGM-G is invoked.

Unlike for other algorithms, we will not be able to argue that $\mathcal{C}_k - \mathcal{C}_{k-1} \le E_k$ for E_k that is either zero or only contains telescoping terms. Instead, we will settle for a more modest goal of arguing that, under the appropriate choice of algorithm steps and growth of the sequence A_k, we have $\mathcal{C}_K \le \mathcal{C}_0$. Observe that, by the definition of $\mathcal{C}_k$, if we can prove that $A_K/A_0 = \Omega(K^2)$, this condition immediately leads to the desired bound

$$\|\nabla f(\boldsymbol{x}_K)\|^2 = O\Big(\frac{L(f(\boldsymbol{x}_0) - f(\boldsymbol{x}_K))}{K^2}\Big) = O\Big(\frac{L(f(\boldsymbol{x}_0) - f(\boldsymbol{x}^*))}{K^2}\Big).$$

As before, we define $a_k = A_k - A_{k-1}$ and assume it is strictly positive, $\forall k$ (i.e., A_k is strictly increasing). To bound $\mathcal{C}_K$, we start by bounding the change in the potential function $\mathcal{C}_k - \mathcal{C}_{k-1}$, for $k \ge 1$, in the following lemma. Observe that the lemma itself is algorithm-independent.

LEMMA 2.5. *Let $\mathcal{C}_k$ be defined by (2.10) $\forall k \in \{0, 1, \dots, K\}$. Define $\boldsymbol{y}_k = \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)$ for $k \ge 0$, and set $\boldsymbol{y}_{-1} = \boldsymbol{x}_0$. Then, $\forall 1 \le k \le K$*

$$\begin{aligned}\mathcal{C}_k - \mathcal{C}_{k-1} \le{}& A_k\left\langle\nabla f(\boldsymbol{x}_k), \boldsymbol{x}_k - \boldsymbol{y}_{k-1}\right\rangle - A_{k-1}\left\langle\nabla f(\boldsymbol{x}_{k-1}), \boldsymbol{x}_{k-1} - \boldsymbol{y}_{k-2}\right\rangle \\ &+ \left\langle\nabla f(\boldsymbol{x}_{k-1}), A_k\boldsymbol{y}_{k-1} - A_{k-1}\boldsymbol{y}_{k-2} - a_k\boldsymbol{y}_K\right\rangle.\end{aligned}$$

Proof. Let $\boldsymbol{x}, \hat{\boldsymbol{x}}$ be any two vectors from $\mathbb{R}^d$, and let $\boldsymbol{y} = \boldsymbol{x} - \frac{1}{L}\nabla f(\boldsymbol{x})$. Then, (1.3) can be equivalently written as

$$f(\hat{\boldsymbol{x}}) - f(\boldsymbol{x}) \leq \langle \nabla f(\hat{\boldsymbol{x}}), \hat{\boldsymbol{x}} - \boldsymbol{y} \rangle - \frac{1}{2L}\|\nabla f(\hat{\boldsymbol{x}})\|^2 - \frac{1}{2L}\|\nabla f(\boldsymbol{x})\|^2. \tag{2.11}$$

From the definition of $\mathcal{C}_k$ in (2.10), we have

$$\begin{aligned}\mathcal{C}_k - \mathcal{C}_{k-1} = \frac{A_k}{2L}\|\nabla f(\boldsymbol{x}_k)\|^2 - \frac{A_{k-1}}{2L}\|\nabla f(\boldsymbol{x}_{k-1})\|^2 + \frac{a_k}{2L}\|\nabla f(\boldsymbol{x}_K)\|^2 \\ + A_k(f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k-1})) + a_k(f(\boldsymbol{x}_{k-1}) - f(\boldsymbol{x}_K)).\end{aligned}$$

Applying (2.11) to $f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k-1})$ and $f(\boldsymbol{x}_{k-1}) - f(\boldsymbol{x}_K)$, we further have

$$\begin{aligned}\mathcal{C}_k - \mathcal{C}_{k-1} \leq & -\frac{A_k}{L}\|\nabla f(\boldsymbol{x}_{k-1})\|^2 + A_k \left\langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x}_k - \boldsymbol{y}_{k-1} \right\rangle \\ & + a_k \left\langle \nabla f(\boldsymbol{x}_{k-1}), \boldsymbol{x}_{k-1} - \boldsymbol{y}_K \right\rangle \\ = & A_k \left\langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x}_k - \boldsymbol{y}_{k-1} \right\rangle + A_k \left\langle \nabla f(\boldsymbol{x}_{k-1}), \boldsymbol{y}_{k-1} - \boldsymbol{y}_K \right\rangle \\ & - A_{k-1} \left\langle \nabla f(\boldsymbol{x}_{k-1}), \boldsymbol{x}_{k-1} - \boldsymbol{y}_K \right\rangle \\ = & A_k \left\langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x}_k - \boldsymbol{y}_{k-1} \right\rangle - A_{k-1} \left\langle \nabla f(\boldsymbol{x}_{k-1}), \boldsymbol{x}_{k-1} - \boldsymbol{y}_{k-2} \right\rangle \\ & + \left\langle \nabla f(\boldsymbol{x}_{k-1}), A_k \boldsymbol{y}_{k-1} - A_{k-1}\boldsymbol{y}_{k-2} - a_k \boldsymbol{y}_K \right\rangle,\end{aligned}$$

as claimed. ∎

The following lemma provides the restrictions on the step sizes of the algorithm that are needed to ensure that $\mathcal{C}_K \leq \mathcal{C}_0$. Here, we assume that each point $\boldsymbol{x}_k$ can be expressed as the sum of the initial point $\boldsymbol{x}_0$ and some linear combination of the gradients evaluated at points $\boldsymbol{x}_i$ for $0 \leq i \leq k-1$. Note that most of the standard first-order algorithms can be expressed in this form.

LEMMA 2.6. *Let $\mathcal{C}_k$ be defined by (2.10) for $k \in \{0, \ldots, K\}$ and assume that points $\boldsymbol{x}_k$ can be expressed as $\boldsymbol{x}_k = \boldsymbol{x}_0 - \frac{1}{L}\sum_{i=0}^{k-1} \beta_{i,k} \nabla f(\boldsymbol{x}_i)$, where $\beta_{i,k}$ are some real scalars. Define $\beta_{k,k} = 1$, so that $\boldsymbol{y}_k = \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k) = \boldsymbol{x}_0 - \frac{1}{L}\sum_{i=0}^{k} \beta_{i,k} \nabla f(\boldsymbol{x}_i)$ and set $\boldsymbol{y}_{-1} = \boldsymbol{x}_0$. If the following two conditions are satisfied for all $0 \leq j < k \leq K-1$,*

$$\beta_{k,K-1} + \frac{a_{k+1}}{A_K} \leq \frac{A_{k+1}}{a_{k+1}}, \tag{2.12}$$

$$A_{k+1}\beta_{j,k} = A_k \beta_{j,k-1} + a_{k+1}\Big(\beta_{j,K-1} + \frac{a_{j+1}}{A_K}\Big) + a_{j+1}\Big(\beta_{k,K-1} + \frac{a_{k+1}}{A_K}\Big), \tag{2.13}$$

and if

$$\boldsymbol{x}_K = \boldsymbol{y}_{K-1} - \frac{1}{LA_K}\sum_{k=0}^{K-1} a_{k+1}\nabla f(\boldsymbol{x}_k), \tag{2.14}$$

then $\mathcal{C}_K \leq \mathcal{C}_0$. Further, the largest growth of $\frac{A_K}{A_0}$ for which both of these conditions can be satisfied is $O(K^2)$.

Proof. Telescoping the inequality from Lemma 2.5, we have

$$\begin{aligned}\mathcal{C}_K - \mathcal{C}_0 \leq & A_K \left\langle \nabla f(\boldsymbol{x}_K), \boldsymbol{x}_K - \boldsymbol{y}_{K-1} \right\rangle \\ & + \sum_{k=0}^{K-1} \left\langle \nabla f(\boldsymbol{x}_k), A_{k+1}\boldsymbol{y}_k - A_k \boldsymbol{y}_{k-1} - a_{k+1}\boldsymbol{y}_K \right\rangle.\end{aligned}$$

Observe that $\nabla f(\boldsymbol{x}_K)$ only appears in the first term and as part of $\boldsymbol{y}_K = \boldsymbol{x}_K - \frac{1}{L}\nabla f(\boldsymbol{x}_K)$. Thus, grouping the terms that multiply $\nabla f(\boldsymbol{x}_K)$, we can, equivalently, write

$$\mathcal{C}_K - \mathcal{C}_0 \leq \left\langle \nabla f(\boldsymbol{x}_K), A_K(\boldsymbol{x}_K - \boldsymbol{y}_{K-1}) + \frac{1}{L}\sum_{k=0}^{K-1} a_{k+1}\nabla f(\boldsymbol{x}_k) \right\rangle + \sum_{k=0}^{K-1} \left\langle \nabla f(\boldsymbol{x}_k), A_{k+1}\boldsymbol{y}_k - A_k\boldsymbol{y}_{k-1} - a_{k+1}\boldsymbol{x}_K \right\rangle.$$

The choice of $\boldsymbol{x}_K$ from (2.14) ensures that the first term on the right-hand side is zero (and this is how it was chosen). The rest of the terms can be expressed as a function of gradients up to the $(K-1)$th one. To simplify the notation, let us define $\boldsymbol{g}_{K-1} = \frac{1}{L}\sum_{k=0}^{K-1} a_{k+1}\nabla f(\boldsymbol{x}_k)$. Then, we have

$$\mathcal{C}_K - \mathcal{C}_0 \leq \sum_{k=0}^{K-1} \left\langle \nabla f(\boldsymbol{x}_k), A_{k+1}\boldsymbol{y}_k - A_k\boldsymbol{y}_{k-1} - a_{k+1}\Big(\boldsymbol{y}_{K-1} - \frac{\boldsymbol{g}_{K-1}}{A_K}\Big) \right\rangle. \tag{2.15}$$

Observe that, as $\boldsymbol{y}_k = \boldsymbol{x}_0 - \frac{1}{L}\sum_{i=0}^{k}\beta_{i,k}\nabla f(\boldsymbol{x}_i)$ by the lemma assumptions, the expression on the right-hand side can be written as a linear combination of inner products between gradients, as follows:

$$\mathcal{C}_K - \mathcal{C}_0 \leq \frac{1}{L}\sum_{j=0}^{K-1}\sum_{k=j}^{K-1} P_{j,k}\left\langle \nabla f(\boldsymbol{x}_j), \nabla f(\boldsymbol{x}_k)\right\rangle,$$

where, by (2.15), we have that, $\forall\, 0 \leq j < k \leq K-1$,

$$P_{k,k} = -A_{k+1}\beta_{k,k} + a_{k+1}\Big(\beta_{k,K-1} + \frac{a_{k+1}}{A_K}\Big),$$
$$P_{j,k} = -A_{k+1}\beta_{j,k} + A_k\beta_{j,k-1} + a_{k+1}\Big(\beta_{j,K-1} + \frac{a_{j+1}}{A_K}\Big) + a_{j+1}\Big(\beta_{k,K-1} + \frac{a_{k+1}}{A_K}\Big).$$

As, by assumption, $\beta_{k,k} = 1$, conditions in (2.12) and (2.13) are equivalent to $P_{k,k} \leq 0$ and $P_{j,k} = 0\,\forall\, 0 \leq j < k \leq K-1$. By construction, these conditions are sufficient for guaranteeing $\mathcal{C}_K - \mathcal{C}_0 \leq 0$, completing the first part of the proof.

Observe that, given a sequence of positive numbers $\{a_k\}_{k\geq 0}$ and $A_k = \sum_{j=0}^{k} a_j$, all coefficients $\beta_{j,k}$ are uniquely determined by (2.13) (as $\beta_{k,k} = 1$ by assumption, and the remaining coefficients can be computed by recursively applying (2.13)). Thus, the role of the condition from (2.12) is to limit the growth of the sequence $\{A_k\}_{k\geq 0}$. Starting with $\beta_{k,k} = 1\ \forall k$ (which holds by assumption), it is possible to argue by induction that $\beta_{j,k} \geq 0\ \forall j,k$ (the proof is omitted for brevity). Thus the condition from (2.12) implies that $\frac{a_{k+1}}{A_K} \leq \frac{A_{k+1}}{a_{k+1}}$. Equivalently, $\forall k \leq K-1$,

$$\frac{{a_{k+1}}^2}{A_{k+1}} \leq A_K. \tag{2.16}$$

For any fixed A_K, (2.16) implies that $\frac{A_k}{A_0}$ cannot grow faster than quadratically with k for $k \leq K-1$. It remains to argue that the sequence does not make a big jump from A_{K-1} to A_K. This follows by using again (2.12) for $k = K-1$ and recalling that $\beta_{K-1,K-1} = 1$. We then have

$$1 + \frac{a_K}{A_K} \leq \frac{A_K}{a_K}.$$

Solving for $\frac{a_K}{A_K}$, it follows that $\frac{a_K}{A_K} \leq \frac{-1+\sqrt{5}}{2} < 0.62$, and thus, $\frac{A_K}{A_{K-1}} \leq \frac{1}{1-0.62} < 3$, completing the proof that $\frac{A_K}{A_0} = O(K^2)$. ∎

That $\frac{A_K}{A_0} = O(K^2)$ is not surprising—if it were not true, by the discussion from the beginning of this subsection, we would be able to obtain an algorithm that converges at rate faster than $1/K^2$, which is impossible, due to the existing lower bounds [15, 47, 48]. This result was included to highlight the role of the conditions from (2.12) and (2.13) in Lemma 2.6: the first condition limits the growth of $\{A_k\}_{k\geq 0}$, whereas the second determines the step sizes $\beta_{j,k}$ in the algorithm, given the sequence $\{A_k\}_{k\geq 0}$.

What remains to be shown is that there is a choice of step sizes $\beta_{j,k}$ that guarantees $\frac{A_K}{A_0} = \Theta(K^2)$, and thus leads to an algorithm with the optimal convergence rate. As we show next, such a choice of $\beta_{j,k}$ can be obtained when the inequality from (2.12) is satisfied with equality. Further, when (2.12) is satisfied with equality, (2.13) can be further simplified, and it leads to the algorithm description that does not necessitate storing all of the gradients, but only a constant number of d-dimensional vectors. However, similar to the algorithm description in [32], the entire sequence $\{A_k\}_{k=0}^K$ needs to be precomputed and stored, which appears to be unavoidable. The algorithm and its convergence rate are summarized in the following theorem.

THEOREM 2.7 (convergence of optimized gradient method). *Let $f : \mathbb{R}^d \to \mathbb{R}$ be an L-smooth function, and let $\boldsymbol{x}_0 \in \mathbb{R}^d$ be an arbitrary initial point. Let $K \geq 1$. Consider the following algorithm. Let $\boldsymbol{v}_0 = \boldsymbol{x}_0 - \frac{A_1}{a_1 L}\nabla f(\boldsymbol{x}_0)$, $\boldsymbol{g}_0 = \frac{a_1}{L}\nabla f(x_0)$. For $k = 1$ to $K-1$,*

$$\begin{aligned} \boldsymbol{y}_{k-1} &= \boldsymbol{x}_{k-1} - \frac{1}{L}\nabla f(\boldsymbol{x}_{k-1}), \\ \boldsymbol{x}_k &= \frac{A_k}{A_{k+1}}\boldsymbol{y}_{k-1} + \frac{a_{k+1}}{A_{k+1}}\boldsymbol{v}_{k-1} - \frac{1}{a_{k+1}}\boldsymbol{g}_{k-1}, \\ \boldsymbol{v}_k &= \boldsymbol{v}_{k-1} - \frac{1}{L}\frac{A_{k+1}}{a_{k+1}}\nabla f(\boldsymbol{x}_k), \quad \boldsymbol{g}_k = \boldsymbol{g}_{k-1} + \frac{a_{k+1}}{L}\nabla f(\boldsymbol{x}_k), \end{aligned} \tag{2.17}$$

where the sequence $\{A_k\}_{k=0}^K$ is recursively defined by the following:

$$\begin{cases} A_k = 1 & \text{if } k = K, \\ A_k = A_{k+1}\left[1 + \frac{1}{2}A_{k+1} - \frac{1}{2}\sqrt{A_{k+1}(4 + A_{k+1})}\right] & \text{if } 0 \leq k \leq K-1, \end{cases} \tag{2.18}$$

and $a_{k+1} = A_{k+1} - A_k$ for $0 \leq k \leq K-1$.

If $\boldsymbol{x}_K$ is defined by

$$\boldsymbol{x}_K = \boldsymbol{y}_{K-1} - \frac{1}{A_K L}\boldsymbol{g}_{K-1},$$

then

$$\|\nabla f(\boldsymbol{x}_K)\|^2 \leq \frac{16L(f(\boldsymbol{x}_0) - f(\boldsymbol{x}^*))}{(K+2)^2},$$

where $\boldsymbol{x}^* \in \operatorname{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$.

Proof. The proof strategy is as follows. We first argue that the algorithm from the theorem statement satisfies $\mathcal{C}_K \leq \mathcal{C}_0$, where $\mathcal{C}_k$ is defined by (2.10). This is done by showing that we can apply Lemma 2.6. Then, by the definition of $\mathcal{C}_k$, $\mathcal{C}_K \leq \mathcal{C}_0$ is equivalent to

$$\|\nabla f(\boldsymbol{x}_K)\|^2 \leq 2L\frac{A_0}{A_K}\left(f(\boldsymbol{x}_0) - f(\boldsymbol{x}_K) + \frac{1}{2L}\|\nabla f(\boldsymbol{x}_0)\|^2\right).$$

As $f(\boldsymbol{x}_K) \geq f(\boldsymbol{x}^*)$ and $\frac{1}{2L}\|\nabla f(\boldsymbol{x}_0)\|^2 \leq f(\boldsymbol{x}_0) - f(\boldsymbol{x}^*)$, what then remains to be argued is that $\frac{A_0}{A_K} = O(\frac{1}{K^2})$.

To apply Lemma 2.6, observe first that the definition of $\boldsymbol{x}_K$ from Theorem 2.7 is the same as the definition of $\boldsymbol{x}_K$ in Lemma 2.6. For $k \leq K-1$, let us define $\boldsymbol{x}_k = \boldsymbol{x}_0 - \frac{1}{L}\sum_{j=0}^{k-1} \beta_{j,k}\nabla f(\boldsymbol{x}_j)$, $\beta_{k,k} = 1$, and $\boldsymbol{y}_k = \boldsymbol{x}_k - \frac{\beta_{k,k}}{L}\nabla f(\boldsymbol{x}_k)$ as in Lemma 2.6 and show that when both conditions from Lemma 2.6 stated in (2.12) and (2.13) are satisfied with equality, we recover the algorithm from the theorem statement, and thus the two sequences of points are equivalent, and so we can conclude that $\mathcal{C}_K \leq \mathcal{C}_0$.

When (2.12) holds with equality, we have that

$$\beta_{k,K-1} + \frac{a_{k+1}}{A_K} = \frac{A_{k+1}}{a_{k+1}}. \tag{2.19}$$

Plugging it into (2.13), we have

$$A_{k+1}\beta_{j,k} = A_k\beta_{j,k-1} + a_{k+1}\frac{A_{j+1}}{a_{j+1}} + a_{j+1}\frac{A_{k+1}}{a_{k+1}}. \tag{2.20}$$

Thus, it follows that

$$\begin{aligned} A_{k+1}\boldsymbol{x}_k - A_k\boldsymbol{y}_{k-1} &= a_{k+1}\boldsymbol{x}_0 - \frac{a_{k+1}}{L}\sum_{j=0}^{k-1}\frac{A_{j+1}}{a_{j+1}}\nabla f(\boldsymbol{x}_j) - \frac{A_{k+1}}{a_{k+1}L}\sum_{j=0}^{k-1} a_{j+1}\nabla f(\boldsymbol{x}_j) \\ &= a_{k+1}\boldsymbol{v}_{k-1} - \frac{A_{k+1}}{a_{k+1}}\boldsymbol{g}_{k-1}, \end{aligned}$$

which is the same as the definition of $\boldsymbol{x}_k$ from (2.17).

It remains to show that the conditions from Lemma 2.6 imply the recursive definition of the sequence $\{A_k\}_{k\geq 0}$ and that $\frac{A_K}{A_0} \geq \frac{4}{(K+2)^2}$. This is established by Lemma A.1 in the appendix. ∎

Remark 2.8. While OGM-G provides the optimal convergence guarantee for norm of the gradient, its convergence rate for the optimality gap is not known. Thus, it does not immediately imply a bound on norm of the gradient in terms of $\|\boldsymbol{x}^* - \boldsymbol{x}_0\|^2$. However, as observed in [53], it is possible to obtain a bound of $\|\nabla f(\boldsymbol{x}_K)\|^2 = O\big(\frac{L^2\|\boldsymbol{x}^*-\boldsymbol{x}_0\|^2}{K^4}\big)$ from OGM-G, by running Nesterov FGM for $\lfloor K/2 \rfloor$ iterations, followed by $\lceil K/2 \rceil$ iterations of OGM-G.

2.3. Discussion. Gradient descent is perhaps the simplest method that can be used for minimizing the gradient norm. We also conjecture that it is, in a certain sense, optimal.

CONJECTURE 2.9. *For any $K > 0$ and any method that constructs its iterates as $\boldsymbol{x}_k = \boldsymbol{x}_0 - \sum_{i=0}^{k-1}\beta_{i,k}\nabla f(\boldsymbol{x}_i)$, where $\boldsymbol{x}_0 \in \mathbb{R}^d$ is the initial point, f is a convex function accessed via a gradient oracle, and coefficients $\beta_{i,k} \in \mathbb{R}$ can depend on $L > 0, i, k$ but are otherwise chosen independently of K or the input function f, there exist an L-smooth convex input function f and an absolute constant $C > 0$ such that*

$$\|\nabla f(\boldsymbol{x}_K)\|^2 \geq C\frac{L(f(\boldsymbol{x}_0) - f(\boldsymbol{x}^*))}{K}.$$

The basis for this conjecture is the numerical evidence from [31, 32], which seems to suggest that fixing the total number of iterations K and choosing the coefficients $\beta_{i,k}$ as a function K is crucial to obtaining the optimal bound $\|\nabla f(\boldsymbol{x}_K)\|^2 =$

$O\big(\frac{L(f(\boldsymbol{x}_0)-f(\boldsymbol{x}^*))}{K^2}\big)$. In particular, using the PEP framework to optimize the coefficients of generalized OGM (GOGM) methods, which are, roughly, two-step or momentum-like methods, only led to the $O(L^2\|\boldsymbol{x}^* - \boldsymbol{x}_0\|^2/k^3)$ bound on the minimum squared gradient norm in [31, section 5]. The resulting method, OGM-OG, also has the following guarantee for the optimality gap: $f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*) = O(L\|\boldsymbol{x}^* - \boldsymbol{x}_0\|^2/k^2)$, and, thus, it cannot obtain a better guarantee for the squared gradient norm than stated in Conjecture 2.9. Note also that, since Nesterov FGM belongs to the class of GOGM methods considered in [31], if there were a hypothetical GOGM with a convergence bound $\|\nabla f(\boldsymbol{x}_k)\|^2 = O(\frac{L(f(\boldsymbol{x}_0)-f(\boldsymbol{x}^*))}{k^2})$, then it would have been possible to obtain a GOGM with the $\|\nabla f(\boldsymbol{x}_k)\|^2 = O(\frac{L^2\|\boldsymbol{x}^*-\boldsymbol{x}_0\|^2}{k^4})$ guarantee, by simply appending half the iterations of the hypothetical GOGM to Nesterov FGM. Hence, this bound appears to be impossible for the class of GOGM methods, which one would expect to be a natural candidate for solving this problem. On the other hand, fixing the number of iterations in the OGM-G method from [32] was crucial for obtaining the $\|\nabla f(\boldsymbol{x}_K)\|^2 = O(\frac{L(f(\boldsymbol{x}_0)-f(\boldsymbol{x}^*))}{K^2})$ bound, and it is unclear whether and how the same bound could be obtained without this requirement.

We note that the lower bound from Conjecture 2.9 can be proved under a stricter condition on coefficients $\beta_{i,k}$ that essentially forces them to be constant (independent of i and k), using the techniques of Arjevani and Shamir [3]. However, such a lower bound is weak as it not only excludes the optimal algorithm from [32] (which is desired) but also all variants of Nesterov FGM considered in [31].

3. Small gradients in min-max optimization. In this section, we consider the problem of making the gradients small in convex-concave min-max optimization, under the assumption that the operator F corresponding to the gradient of the objective is cocoercive (see section 1.2). Similarly, as in the case of convex optimization, the potential functions we consider trade off a notion of an optimality gap with the norm of F. Further, the inequality corresponding to the cocoercivity assumption suffices to carry out the analysis of standard methods considered here; namely, the gradient descent-ascent method and Halpern iteration. We also show (in section 3.3) that these two methods are the best we can hope for when considering broad classes of methods that capture most of the standard optimization methods.

3.1. Krasnosel'skiı–Mann/gradient descent-ascent. Perhaps the simplest potential function that can be considered for min-max optimization is

$$\mathcal{C}_k = A_k\|F(\boldsymbol{u}_k)\|^2 + B_k\left\langle F(\boldsymbol{u}_k), \boldsymbol{u}_k - \boldsymbol{u}^*\right\rangle, \tag{3.1}$$

which can be seen as a counterpart to the potential function used for gradient descent in the previous section. The method that is suitable for the analysis with this potential function is also the counterpart of gradient descent for min-max optimization—gradient descent-ascent (GDA), stated as

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k - \eta_k F(\boldsymbol{u}_k),$$

where $\eta_k \in (0, \frac{2}{L})$. This method is also equivalent to the well-known Krasnosel'skiı–Mann iteration for finding fixed points of nonexpansive (1-Lipschitz) operators. In particular, given a nonexpansive operator $T : \mathbb{R}^d \to \mathbb{R}^d$, the Krasnosel'skiı–Mann iteration updates the iterates as

$$\boldsymbol{u}_{k+1} = (1-\alpha_k)\boldsymbol{u}_k + \alpha_k T(\boldsymbol{u}_k),$$

where $\alpha_k \in (0,1)$. It is a standard fact that F is $\frac{1}{L}$-cocoercive if and only if $T(\cdot) = \cdot - \frac{2}{L}F(\cdot)$ is nonexpansive (see, e.g., [10, Proposition 4.1]). Thus, if we apply the Krasnosel'skiı–Mann iteration to $T(\cdot) = \cdot - \frac{2}{L}F(\cdot)$, we have

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k - \frac{2\alpha_k}{L}F(\boldsymbol{u}_k),$$

which is precisely GDA with $\eta_k = \frac{2\alpha_k}{L}$.

For simplicity, in the following we analyze GDA with the step size $\eta_k = \eta = \frac{1}{L}$, which is the optimal step size for this method. The analysis, however, extends to any step sizes $\eta_k \in (0, \frac{2}{L})$ in a straightforward manner. The convergence result is summarized in the following lemma. We note that, similar to other convergence results in this paper, this is a well-known result [13, 26, 59], and it is only the potential function-based argument that is new.

LEMMA 3.1 (convergence of GDA). *Let $F : \mathbb{R}^d \to \mathbb{R}^d$ be a $\frac{1}{L}$-cocoercive operator, let $\boldsymbol{u}_0 \in \mathbb{R}^d$ be an arbitrary initial point, and let $\boldsymbol{u}_{k+1} = \boldsymbol{u}_k - \frac{1}{L}F(\boldsymbol{u}_k)$ for $k \geq 0$. Then $\forall k \geq 1$,*

$$\|F(\boldsymbol{u}_k)\| \leq \frac{L\|\boldsymbol{u}_0 - \boldsymbol{u}^*\|}{\sqrt{k+1}},$$

where $\boldsymbol{u}^*$ is such that $F(\boldsymbol{u}^*) = \boldsymbol{0}$.

Proof. The proof relies on showing that the potential function $\mathcal{C}_k$ satisfies $\mathcal{C}_k \leq \mathcal{C}_{k-1} + E_k$, where E_k only contains terms that telescope for suitably chosen sequences of positive numbers $\{A_k\}_{k\geq 0}$ and $\{B_k\}_{k\geq 0}$.

Let us start with bounding $\mathcal{C}_0$. As $\boldsymbol{u}_1 = \boldsymbol{u}_0 - \frac{1}{L}F(\boldsymbol{u}_0)$, we have

$$\begin{aligned}
\mathcal{C}_0 &= A_0\|F(\boldsymbol{u}_0)\|^2 + B_0\langle F(\boldsymbol{u}_0), \boldsymbol{u}_0 - \boldsymbol{u}^*\rangle \\
&= A_0\|F(\boldsymbol{u}_0)\|^2 + B_0L\langle \boldsymbol{u}_0 - \boldsymbol{u}_1, \boldsymbol{u}_0 - \boldsymbol{u}^*\rangle \\
&= A_0\|F(\boldsymbol{u}_0)\|^2 + \frac{B_0L}{2}\big(\|\boldsymbol{u}_0 - \boldsymbol{u}^*\|^2 - \|\boldsymbol{u}_1 - \boldsymbol{u}^*\|^2 + \|\boldsymbol{u}_0 - \boldsymbol{u}_1\|^2\big) \\
&= \Big(A_0 + \frac{B_0}{2L}\Big)\|F(\boldsymbol{u}_0)\|^2 + \frac{B_0L}{2}\big(\|\boldsymbol{u}_0 - \boldsymbol{u}^*\|^2 - \|\boldsymbol{u}_1 - \boldsymbol{u}^*\|^2\big). \qquad (3.2)
\end{aligned}$$

Now let us consider the change in the potential function $\mathcal{C}_k - \mathcal{C}_{k-1}$. Note first that, by (1.7), $\langle F(\boldsymbol{u}_{k-1}), \boldsymbol{u}_{k-1} - \boldsymbol{u}^*\rangle \geq \frac{1}{L}\|F(\boldsymbol{u}_{k-1})\|^2$. Thus,

$$\begin{aligned}
\mathcal{C}_k - \mathcal{C}_{k-1} &= A_k\|F(\boldsymbol{u}_k)\|^2 - A_{k-1}\|F(\boldsymbol{u}_{k-1})\|^2 + B_k\langle F(\boldsymbol{u}_k), \boldsymbol{u}_k - \boldsymbol{u}^*\rangle \\
&\quad - B_{k-1}\langle F(\boldsymbol{u}_{k-1}), \boldsymbol{u}_{k-1} - \boldsymbol{u}^*\rangle \\
&\leq A_k\|F(\boldsymbol{u}_k)\|^2 - \Big(A_{k-1} + \frac{B_{k-1}}{L}\Big)\|F(\boldsymbol{u}_{k-1})\|^2 + B_k\langle F(\boldsymbol{u}_k), \boldsymbol{u}_k - \boldsymbol{u}^*\rangle.
\end{aligned}$$

Using that $F(\boldsymbol{u}_k) = L(\boldsymbol{u}_k - \boldsymbol{u}_{k+1})$, we have that $\langle F(\boldsymbol{u}_k), \boldsymbol{u}_k - \boldsymbol{u}^*\rangle = \frac{1}{2L}\|F(\boldsymbol{u}_k)\|^2 + \frac{L}{2}\|\boldsymbol{u}_k - \boldsymbol{u}^*\|^2 - \frac{L}{2}\|\boldsymbol{u}_{k+1} - \boldsymbol{u}^*\|^2$, which leads to

$$\begin{aligned}
\mathcal{C}_k - \mathcal{C}_{k-1} \leq \Big(A_k + \frac{B_k}{2L}\Big)\|F(\boldsymbol{u}_k)\|^2 - \Big(A_{k-1} + \frac{B_{k-1}}{L}\Big)\|F(\boldsymbol{u}_{k-1})\|^2 \\
+ \frac{B_kL}{2}\|\boldsymbol{u}_k - \boldsymbol{u}^*\|^2 - \frac{B_kL}{2}\|\boldsymbol{u}_{k+1} - \boldsymbol{u}^*\|^2.
\end{aligned}$$

On the other hand, by (1.6) and $\boldsymbol{u}_k = \boldsymbol{u}_{k-1} - \frac{1}{L}F(\boldsymbol{u}_{k-1})$, we have that $\|F(\boldsymbol{u}_k)\|^2 \leq \langle F(\boldsymbol{u}_k), F(\boldsymbol{u}_{k-1})\rangle$, and consequently, $\|F(\boldsymbol{u}_k)\| \leq \|F(\boldsymbol{u}_{k-1})\|$. Thus, for $\mathcal{C}_k - \mathcal{C}_{k-1}$ to

contain only telescoping terms, it suffices that $A_k + \frac{B_k}{2L} - A_{k-1} - \frac{B_{k-1}}{L} \leq 0$ and that $\{B_k\}_{k\geq 0}$ is nonincreasing. In particular, taking $B_k = 1$, we have

$$\mathcal{C}_k - \mathcal{C}_{k-1} \leq \frac{L}{2}\|\boldsymbol{u}_k - \boldsymbol{u}^*\|^2 - \frac{L}{2}\|\boldsymbol{u}_{k+1} - \boldsymbol{u}^*\|^2 - \Big(A_{k-1} - A_k + \frac{1}{2L}\Big)\|F(\boldsymbol{u}_{k-1})\|^2. \tag{3.3}$$

Telescoping (3.3), combining with (3.2), and choosing $A_0 = A_1 = 0$, $A_k = A_{k-1} + \frac{1}{2L}$ for $k \geq 2$, we then get

$$\mathcal{C}_k \leq \frac{L}{2}\|\boldsymbol{u}_0 - \boldsymbol{u}^*\|^2 - \frac{L}{2}\|\boldsymbol{u}_{k+1} - \boldsymbol{u}^*\|^2 \leq \frac{L}{2}\|\boldsymbol{u}_0 - \boldsymbol{u}^*\|^2.$$

Finally, observing that, by (1.7), $\mathcal{C}_k \geq \big(A_k + \frac{B_k}{L}\big)\|F(\boldsymbol{u}_k)\|^2 = \frac{k+1}{2L}\|F(\boldsymbol{u}_k)\|^2$ for all $k \geq 1$, we finally get

$$\|F(\boldsymbol{u}_k)\|^2 \leq \frac{L^2\|\boldsymbol{u}_0 - \boldsymbol{u}^*\|^2}{k+1}.$$

It remains to take the square-root on both sides of the last inequality. ∎

3.2. Halpern iteration. It seems reasonable now to ask whether it is possible to obtain faster rates than for GDA by considering a different potential function that trades off the gradient/operator norm for a notion of an optimality gap w.r.t. an anchor point, similar to how we obtained faster rates for convex optimization. It turns out that the answer is "yes," using the initial point $\boldsymbol{u}_0$ as the anchor. The resulting potential function is

$$\mathcal{C}_k = A_k\|F(\boldsymbol{u}_k)\|^2 + B_k\,\langle F(\boldsymbol{u}_k), \boldsymbol{u}_k - \boldsymbol{u}_0\rangle \tag{3.4}$$

and it corresponds to the well-known Halpern iteration

$$\boldsymbol{u}_{k+1} = \lambda_{k+1}\boldsymbol{u}_0 + (1 - \lambda_{k+1})T(\boldsymbol{u}_k), \tag{3.5}$$

where, similarly as in the case of GDA, $T(\cdot) = \cdot - \frac{2}{L}F(\cdot)$ is a nonexpansive operator. We note that a similar potential function was used in [17] to analyze the convergence of Halpern iteration.

The main convergence result is summarized in the following lemma. While the same convergence result was proved in [38, Thereom 2.1] using the PEP framework [22], the potential function-based argument provided here directly leads to Halpern iteration by enforcing the condition that the potential function defined by (3.4) is nonincreasing.

LEMMA 3.2 (convergence of Halpern iteration). *Let $F : \mathbb{R}^d \to \mathbb{R}^d$ be a $\frac{1}{L}$-cocoercive operator, let $\boldsymbol{u}_0 \in \mathbb{R}^d$ be an arbitrary initial point, and, for $k \geq 0$, let*

$$\boldsymbol{u}_{k+1} = \frac{1}{k+1}\boldsymbol{u}_0 + \frac{k}{k+1}\Big(\boldsymbol{u}_k - \frac{2}{L}F(\boldsymbol{u}_k)\Big).$$

Then, $\forall k \geq 1$ we have

$$\|F(\boldsymbol{u}_k)\| \leq \frac{L\|\boldsymbol{u}_0 - \boldsymbol{u}^*\|}{k+1},$$

where $\boldsymbol{u}^*$ satisfies $F(\boldsymbol{u}^*) = \boldsymbol{0}$.

Proof. The claim trivially holds if $\|F(\boldsymbol{u}_k)\| = 0$, so assume throughout that $\|F(\boldsymbol{u}_k)\| \neq 0$.

Consider bounding $\mathcal{C}_k - \mathcal{C}_{k-1}$ above by zero. To do so, we can only rely on cocoercivity of F from (1.6). Applying (1.6) with $\boldsymbol{u} = \boldsymbol{u}_k$ and $\boldsymbol{v} = \boldsymbol{u}_{k-1}$ and rearranging the terms, we have

$$\begin{aligned}\frac{1}{L}\|F(\boldsymbol{u}_k)\|^2 \le \Big\langle F(\boldsymbol{u}_k), \boldsymbol{u}_k - \boldsymbol{u}_{k-1} + \frac{2}{L}F(\boldsymbol{u}_{k-1})\Big\rangle \\ - \langle F(\boldsymbol{u}_{k-1}), \boldsymbol{u}_k - \boldsymbol{u}_{k-1}\rangle - \frac{1}{L}\|F(\boldsymbol{u}_{k-1})\|^2.\end{aligned} \tag{3.6}$$

Combining (3.6) with the definition of $\mathcal{C}_k$ and grouping appropriate terms, we have

$$\begin{aligned}\mathcal{C}_k - \mathcal{C}_{k-1} \le \Big\langle F(\boldsymbol{u}_k), A_k L\Big(\boldsymbol{u}_k - \boldsymbol{u}_{k-1} + \frac{2}{L}F(\boldsymbol{u}_{k-1})\Big) + B_k(\boldsymbol{u}_k - \boldsymbol{u}_0)\Big\rangle \\ - \langle F(\boldsymbol{u}_{k-1}), A_k L(\boldsymbol{u}_k - \boldsymbol{u}_{k-1}) + B_{k-1}(\boldsymbol{u}_{k-1} - \boldsymbol{u}_0)\rangle \\ - (A_k + A_{k-1})\|F(\boldsymbol{u}_{k-1})\|^2.\end{aligned} \tag{3.7}$$

To make $\mathcal{C}_k - \mathcal{C}_{k-1}$ nonpositive, it suffices to ensure that the inner-product term from the first line in (3.7) is zero and the sum of remaining terms is nonpositive. The former is achieved by simply setting

$$A_k L\Big(\boldsymbol{u}_k - \boldsymbol{u}_{k-1} + \frac{2}{L}F(\boldsymbol{u}_{k-1})\Big) + B_k(\boldsymbol{u}_k - \boldsymbol{u}_0) = 0. \tag{3.8}$$

For the latter, it suffices that

$$- \langle F(\boldsymbol{u}_{k-1}), A_k L(\boldsymbol{u}_k - \boldsymbol{u}_{k-1}) + B_{k-1}(\boldsymbol{u}_{k-1} - \boldsymbol{u}_0)\rangle - (A_k + A_{k-1})\|F(\boldsymbol{u}_{k-1})\|^2 \le 0. \tag{3.9}$$

Rearranging (3.8) gives the Halpern algorithm from (3.5) with $\lambda_k = \frac{B_k}{A_k L + B_k}$, i.e.,

$$\boldsymbol{u}_k = \frac{B_k}{A_k L + B_k}\boldsymbol{u}_0 + \frac{A_k L}{A_k L + B_k}\Big(\boldsymbol{u}_{k-1} - \frac{2}{L}F(\boldsymbol{u}_{k-1})\Big). \tag{3.10}$$

The other condition (from (3.9)) effectively constrains the growth of A_k compared to B_k, which is expected, as otherwise we would be able to prove an arbitrarily fast convergence rate for Halpern iteration, which is impossible, due to existing lower bounds (see, e.g., [17, Lemma 16(c)]).

On the other hand, (3.9) can be equivalently written as

$$- \langle F(\boldsymbol{u}_{k-1}), A_k L\boldsymbol{u}_k - B_{k-1}\boldsymbol{u}_0 - (A_k L - B_{k-1})\boldsymbol{u}_{k-1}\rangle \le (A_k + A_{k-1})\|F(\boldsymbol{u}_{k-1})\|^2.$$

Now, to guarantee that the last inequality is satisfied and consistent with (3.10), it suffices that

$$\frac{B_{k-1}}{A_k L} = \frac{B_k}{A_k L + B_k} \quad \text{and} \quad \frac{2A_k}{A_k L + B_k} \le \frac{A_k + A_{k-1}}{A_k L}. \tag{3.11}$$

In particular, when $B_k = k + 1$ and $A_k = \frac{k(k+1)}{L}$, both conditions from (3.11) are satisfied with equality.

Hence, for $B_k = k + 1$, $A_k = \frac{k(k+1)}{L}$, and $\lambda_k = \frac{B_k}{A_k L + B_k} = \frac{1}{k+1}$, we have that $\mathcal{C}_k \le \mathcal{C}_0$. By definition, and as $A_0 = 0$, we have that $\mathcal{C}_0 = 0$. Thus, $\mathcal{C}_k \le 0\ \forall k \ge 1$, and it follows that

$$\begin{aligned}\|F(\boldsymbol{u}_k)\|^2 &\le \frac{B_k}{A_k}\langle F(\boldsymbol{u}_k), \boldsymbol{u}_0 - \boldsymbol{u}_k\rangle \\ &= \frac{L}{k}\big(\langle F(\boldsymbol{u}_k), \boldsymbol{u}^* - \boldsymbol{u}_k\rangle + \langle F(\boldsymbol{u}_k), \boldsymbol{u}_0 - \boldsymbol{u}^*\rangle\big) \\ &\le \frac{L}{k}\Big(-\frac{1}{L}\|F(\boldsymbol{u}_k)\|^2 + \|F(\boldsymbol{u}_k)\|\|\boldsymbol{u}_0 - \boldsymbol{u}^*\|\Big),\end{aligned}$$

where the last inequality is by (1.7) and Cauchy–Schwarz. To complete the proof, it remains to rearrange the last inequality and divide both sides by $\|F(\boldsymbol{u}_k)\|$. ∎

3.3. Lower bounds for cocoercive operators. In this section, we provide a lower bound that applies to the class of algorithms that construct their iterates as the sum of an initial point and a linear combination of the cocoercive operator $F : \mathbb{R}^d \to \mathbb{R}^d$ evaluated at any of the points seen up to the current iteration. In particular, given a $\frac{1}{L}$-cocoercive operator $F : \mathbb{R}^d \to \mathbb{R}^d$, an algorithm's iterate $\boldsymbol{u}_k$ at iteration k can be expressed as

$$\boldsymbol{u}_k = \boldsymbol{u}_0 - \sum_{i=0}^{k-1} \beta_{i,k} F(\boldsymbol{u}_i), \tag{3.12}$$

where $\beta_{i,k}$ are real coefficients that can depend on L but are otherwise independent of F. To state the lower bound, we use $\mathcal{F}_{L,D}$ to denote the class of problems with $\frac{1}{L}$-cocoercive operators F that satisfy $\|\boldsymbol{u}^* - \boldsymbol{u}_0\| \leq D$, where $\boldsymbol{u}_0 \in \mathbb{R}^d$ is an arbitrary initial point and $\boldsymbol{u}^*$ is such that $F(\boldsymbol{u}^*) = \boldsymbol{0}$. We assume w.l.o.g. that d is even.

To derive the lower bound, we use the framework developed in [2, 3]. To make use of this framework, which relies on the use of Chebyshev polynomials, it is necessary to construct hard instances corresponding to linear operators $F(\boldsymbol{u}) = \boldsymbol{A}\boldsymbol{u} + \boldsymbol{b}$, where $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ and $\boldsymbol{b} \in \mathbb{R}^d$. We note that such an approach was also used in [25] for the class of monotone Lipschitz operators. However, here we aim to provide a lower bound for the more restricted class of cocoercive operators, which necessitates a separate construction. In particular, the monotone operator from the lower bound instance used in [25] is not cocoercive as it corresponds to a bilinear function; in fact, it satisfies $\langle F(\boldsymbol{u}) - F(\boldsymbol{v}), \boldsymbol{u} - \boldsymbol{v}\rangle = 0\ \forall \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$.

Before delving into the technical details of our lower bound, we first provide definitions and supporting claims from [2] that are needed for stating and proving it. A useful definition is that of 1-SCLI algorithms, which allows abstracting algorithms of the form from (3.12) through the lens of Chebyshev polynomials. Here, we adopt the terminology from [25], which somewhat blurs the lines between various definitions (of stationary, oblivious, p-SCLI) algorithm types from [2, 3], but provides perhaps the simplest way of stating the results.

DEFINITION 3.3 (1-SCLI Algorithms). *An optimization algorithm $\mathcal{A}$ acting on the class of linear operators $F : \mathbb{R}^d \to \mathbb{R}^d$ of the form $F(\boldsymbol{u}) = \boldsymbol{A}\boldsymbol{u}+\boldsymbol{b}$, where $\boldsymbol{A} \in \mathbb{R}^{d\times d}$, $\boldsymbol{b} \in \mathbb{R}^d$, is said to be 1-stationary canonical linear iterative (1-SCLI) over $\mathbb{R}^d$ if, given an initial point $\boldsymbol{u}_0 \in \mathbb{R}^d$, there exist mappings $C_0(\mathbf{A}), N(\mathbf{A}) : \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$ such that for all $k \geq 1$ the iterates of $\mathcal{A}$ can be expressed as*

$$\boldsymbol{u}_k = C_0(\boldsymbol{A})\boldsymbol{u}_{k-1} + N(\boldsymbol{A})\boldsymbol{b}.$$

Observe here that Definition 3.3 imposes no restrictions on what kind of mappings C_0 and N can be. In particular, they can be polynomials of an arbitrary degree. This is important because choosing polynomials of degree K would allow us to emulate arbitrary algorithms of the form from (3.12) run over K iterations, as F is assumed to be linear (this observation is typically used in the analysis of the classical conjugate gradient method; see, e.g., [55, Chapter 5]). On the other hand, restricting the degree of the polynomials would restrict the adaptivity of coefficients $\beta_{i,k}$, as C_0, N remain fixed for all k. In this context, both GDA and Halpern iteration (when restricted to be run over a fixed number K of iterations) can be viewed as 1-SCLI algorithms, with

the following crucial difference. For GDA with a fixed step size η, we have

$$\boldsymbol{u}_k = (\boldsymbol{I} - \eta \boldsymbol{A})\boldsymbol{u}_{k-1} + \eta \boldsymbol{b},$$

i.e., C_0 is of degree one and N is of degree zero. On the other hand, for Halpern iteration,

$$\boldsymbol{u}_k = \lambda_k \boldsymbol{u}_0 + (1-\lambda_k)\Big(\boldsymbol{I} - \frac{2}{L}\boldsymbol{A}\Big)\boldsymbol{u}_{k-1} + (1-\lambda_k)\boldsymbol{b}. \tag{3.13}$$

By recursively applying (3.13) and rolling it down to zero, we get that $\boldsymbol{u}_k$ can be expressed as $\boldsymbol{u}_k = C_0(\boldsymbol{A})\boldsymbol{u}_0 + N(\boldsymbol{A})\boldsymbol{b}$ using C_0 that is a polynomial of degree k and N that is a polynomial of degree $k-1$. In other words, we can view k iterations of Halpern's algorithm as one iteration of a 1-SCLI algorithm, using polynomial maps C_0 and N of suitably large degrees. This is crucial for understanding the statement of the lower bound, which will effectively tell us that GDA is iteration complexity-optimal among all algorithms of the form from (3.12) that choose step sizes $\beta_{i,k}$ independently of k, while Halpern iteration is iteration complexity-optimal over all algorithms that are allowed to adapt $\beta_{i,k}$'s to k.

In the following, we further restrict our attention to operators F corresponding to full-rank matrices $\boldsymbol{A}$. This is convenient because the optimal solution $\boldsymbol{u}^*$ for which $F(\boldsymbol{u}^*) = \boldsymbol{0}$ can be expressed in closed form as $\boldsymbol{u}^* = -\boldsymbol{A}^{-1}\boldsymbol{b}$. This allows us to relate the polynomials C_0 and N under a minimal (and standard [2, 3, 25]) assumption that the 1-SCLI algorithms we consider are *consistent* (or convergent). We note here that the consistency condition is not necessary; it is rather the case that the proof relies on the relationship between C_0 and N from (3.14), for which the natural consistency condition suffices.

DEFINITION 3.4 (consistency). *A 1-SCLI algorithm $\mathcal{A}$ is said to be consistent w.r.t. a full-rank matrix $\boldsymbol{A}$ if for any $\boldsymbol{b} \in \mathbb{R}^d$ we have that $\boldsymbol{u}_k$ converges to $\boldsymbol{u}^* = -\boldsymbol{A}^{-1}\boldsymbol{b}$. A 1-SCLI algorithm is said to be consistent if it is consistent w.r.t. any full-rank matrix $\boldsymbol{A}$.*

The relationship between C_0 and N for consistent algorithms is characterized by the following lemma.

LEMMA 3.5 (consistency of 1-SCLI algorithms [2]). *If a 1-SCLI algorithm is consistent w.r.t. $\boldsymbol{A}$, then*

$$C_0(\boldsymbol{A}) = \boldsymbol{I} + N(\boldsymbol{A})\boldsymbol{A}. \tag{3.14}$$

Finally, the following auxiliary lemma will be useful when proving our lower bound.

LEMMA 3.6 (see [25, Lemma 13]). *Let $L > 0$, let p and k be arbitrary but fixed nonnegative integers, and let $r(y)$ be a polynomial with real-valued coefficients of degree at most p, such that $r(0) = 1$. Then,*

$$\sup_{y \in (0,L]} y|r(y)|^k \geq \sup_{y \in [L/(20p^2k),L]} y|r(y)|^k > \frac{L}{40p^2k}. \tag{3.15}$$

We are now ready to state and prove our lower bound.

THEOREM 3.7. *Let p, K be any two positive integer numbers, and let $L, D > 0$. Then, for any consistent 1-SCLI algorithm $\mathcal{A}$ acting on instances from $\mathcal{F}_{L,D}$, initialized at $\boldsymbol{u}_0 = \boldsymbol{0}$[5] and for which $N(\boldsymbol{A})$ is a matrix polynomial of degree at most $p-1$,*

$$\sup_{F \in \mathcal{F}_{L,D}} \|F(\boldsymbol{u}_K)\| \geq \frac{LD}{4p\sqrt{5K}}.$$

Proof. Similar to [25], we start by showing that

$$\boldsymbol{u}_k = (C_0(\boldsymbol{A})^k - \boldsymbol{I})\boldsymbol{A}^{-1}\boldsymbol{b} \tag{3.16}$$

$\forall\ k \geq 0$. This claim follows by induction on k. The base case $k = 0$ is immediate. For the inductive step, suppose that (3.16) holds for some $k - 1 \geq 0$. Then by the definition of 1-SCLI algorithms and the consistency of $\mathcal{A}$ (Definitions 3.3, 3.4) and Lemma 3.5,

$$\begin{aligned}\boldsymbol{u}_k &= C_0(\boldsymbol{A})\boldsymbol{u}_{k-1} + N(\boldsymbol{A})\boldsymbol{b} \\ &= C_0(\boldsymbol{A})(C_0(\boldsymbol{A})^{k-1} - \boldsymbol{I})\boldsymbol{A}^{-1}\boldsymbol{b} + (C_0(\boldsymbol{A}) - \boldsymbol{I})\boldsymbol{A}^{-1}\boldsymbol{b} \\ &= (C_0(\boldsymbol{A})^k - \boldsymbol{I})\boldsymbol{A}^{-1}\boldsymbol{b}.\end{aligned}$$

Therefore, $F(\boldsymbol{u}_k)$ can be expressed as

$$F(\boldsymbol{u}_k) = \boldsymbol{A}\boldsymbol{u}_k + \boldsymbol{b} = C_0(\boldsymbol{A})^k\boldsymbol{b}. \tag{3.17}$$

Let us now specify the "hard instance." Consider $F(\boldsymbol{u}) = \boldsymbol{A}\boldsymbol{u} + \boldsymbol{b}$, where $\boldsymbol{A}$ can be expressed as $\boldsymbol{A} = \left[\begin{smallmatrix} \eta\boldsymbol{I} & \alpha\boldsymbol{I} \\ -\alpha\boldsymbol{I} & \eta\boldsymbol{I} \end{smallmatrix}\right]$ for some $\eta, \alpha \in \mathbb{R}_+$. (Observe that such an F can be obtained from the convex-concave objective $\phi(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}\eta\boldsymbol{x}^T\boldsymbol{x} - \frac{1}{2}\eta\boldsymbol{y}^T\boldsymbol{y} + \alpha\boldsymbol{x}^T\boldsymbol{y} + \boldsymbol{b}_1^T\boldsymbol{x} - \boldsymbol{b}_2^T\boldsymbol{y}$, where $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{b}_1, \boldsymbol{b}_2 \in \mathbb{R}^{d/2}$, $\boldsymbol{b} = [\boldsymbol{b}_1{}^T \boldsymbol{b}_2{}^T]^T$.)

Let us now argue that for suitably chosen η, α, we have that F is $\frac{1}{L}$-cocoercive. Let $\boldsymbol{u} = [\boldsymbol{x}^T\ \boldsymbol{y}^T]^T$, $\bar{\boldsymbol{u}} = [\bar{\boldsymbol{x}}^T\ \bar{\boldsymbol{y}}^T]^T$ be an arbitrary pair of vectors from $\mathbb{R}^d$, where $\boldsymbol{x}, \boldsymbol{y}, \bar{\boldsymbol{x}}, \bar{\boldsymbol{y}} \in \mathbb{R}^{d/2}$. Then,

$$\langle F(\boldsymbol{u}) - F(\bar{\boldsymbol{u}}), \boldsymbol{u} - \bar{\boldsymbol{u}}\rangle = \eta\|\boldsymbol{u} - \bar{\boldsymbol{u}}\|^2$$

and

$$\|F(\boldsymbol{u}) - F(\bar{\boldsymbol{u}})\|^2 = (\eta^2 + \alpha^2)\|\boldsymbol{u} - \bar{\boldsymbol{u}}\|^2.$$

Hence, for $\eta^2 + \alpha^2 \leq L\eta$, we have $\langle F(\boldsymbol{u}) - F(\bar{\boldsymbol{u}}), \boldsymbol{u} - \bar{\boldsymbol{u}}\rangle \geq \frac{1}{L}\|F(\boldsymbol{u}) - F(\bar{\boldsymbol{u}})\|^2$, i.e., F is $\frac{1}{L}$-cocoercive.

To complete the proof, it remains to show that

$$\sup_{F \in \mathcal{F}_{L,D}} \|F(\boldsymbol{u}_K)\| \geq \frac{LD}{p\sqrt{80K}}.$$

To do so, observe that by (3.17), $\boldsymbol{u}_0 = \boldsymbol{0}$, and $\boldsymbol{u}^* = -\boldsymbol{A}^{-1}\boldsymbol{b}$, we have

$$\sup_{F \in \mathcal{F}_{L,D}} \frac{\|F(\boldsymbol{u}_K)\|^2}{\|\boldsymbol{u}^* - \boldsymbol{u}_0\|^2} \geq \sup_{\substack{\eta \in [0,L], \\ \alpha \in [0, \sqrt{L\eta - \eta^2}]}} \frac{\|C_0(\boldsymbol{A})^K\boldsymbol{b}\|^2}{\|\boldsymbol{A}^{-1}\boldsymbol{b}\|^2},$$

[5] By definition, instances from $\mathcal{F}_{L,D}$ are initialized arbitrarily. However, we can always translate $\boldsymbol{u}$ to $\boldsymbol{u}' = \boldsymbol{u} - \boldsymbol{u}_0$ to make the problem initialized at $\boldsymbol{u}'_0 = 0$.

where $\boldsymbol{A} = \left[\begin{smallmatrix} \eta \boldsymbol{I} & \alpha \boldsymbol{I} \\ -\alpha \boldsymbol{I} & \eta \boldsymbol{I} \end{smallmatrix}\right]$. Observe that the characteristic polynomial of $\boldsymbol{A} = \left[\begin{smallmatrix} \eta \boldsymbol{I} & \alpha \boldsymbol{I} \\ -\alpha \boldsymbol{I} & \eta \boldsymbol{I} \end{smallmatrix}\right]$ is

$$\det(\lambda \boldsymbol{I} - \mathbf{A}) = ((\lambda - \eta)^2 + \alpha^2)^{d/2}.$$

Hence, $\boldsymbol{A}$ has the eigenvalues $\lambda_1 = \eta + \alpha i$, $\lambda_2 = \eta - \alpha i$. These conjugate eigenvalues have the same magnitude: $\sqrt{\eta^2 + \alpha^2}$. Accordingly, $\mathbf{A}^{-1}$ has the eigenvalues $\lambda_1' = \frac{1}{\lambda_1}$, $\lambda_2' = \frac{1}{\lambda_2}$, which are also conjugate and equal in magnitude. On the other hand, since $C_0(\boldsymbol{A}) = \boldsymbol{I} + N(\boldsymbol{A})\boldsymbol{A}$, and, by assumption, $N(\boldsymbol{A})$ is a matrix polynomial of degree at most $p-1$ for some $p \in \mathbb{N}$ with real coefficients, $C_0(\boldsymbol{A})$ is a polynomial of $\boldsymbol{A}$ with $C_0(\mathbf{0}_{d\times d}) = \boldsymbol{I}$. Therefore, it can be expressed as

$$C_0(\boldsymbol{A}) = \boldsymbol{I} + r_1 \boldsymbol{A} + r_2 \mathbf{A}^2 + \cdots + r_p \boldsymbol{A}^p$$

for some real-valued $r_1, r_2, r_3, \ldots, r_p$. We denote the polynomial on complex field with the same real-valued coefficients as $c_0(y) = 1 + r_1 y + r_2 y^2 + \cdots + r_p y^p$. Then, by the spectral mapping theorem, the eigenvalues of $C_0(\boldsymbol{A})$ are $c_0(\lambda_1)$ and $c_0(\lambda_2)$, which are again conjugate and have equal norms. Therefore, we have

$$\begin{aligned} \sup_{\substack{\eta\in[0,L] \\ \alpha\in[0,\sqrt{L\eta-\eta^2}]}} \frac{\left\|C_0(\mathbf{A})^K \boldsymbol{b}\right\|^2}{\left\|\mathbf{A}^{-1}\boldsymbol{b}\right\|^2} &= \sup_{\substack{\eta\in[0,L] \\ \alpha\in[0,\sqrt{L\eta-\eta^2}]}} \frac{|c_0(\lambda_1)|^{2K} \left\|\boldsymbol{b}\right\|^2}{\frac{1}{|\lambda_1|^2}\left\|\boldsymbol{b}\right\|^2} \\ &= \sup_{\substack{\eta\in[0,L] \\ \alpha\in[0,\sqrt{L\eta-\eta^2}]}} (\eta^2+\alpha^2)|c_0(\eta+\alpha i)|^{2K}. \end{aligned}$$

To derive the stated lower bound by applying Lemma 3.6, we need to convert the above expression into a similar form: $\sup_{y\in(0,L]} y|r(y)|^k$. Here, we can observe the difference between the problem we are considering and the problem discussed in [25]. In [25], the eigenvalues are purely imaginary: νi and $-\nu i$. As a result, the above expression can be written as $\sup_{\nu\in(0,L]} \nu^2 |c_0(\nu i)|^{2K}$. By taking the real part of this term, we get a smaller value $\sup_{\nu\in(0,L]} \nu^2 |1 - r_2\nu^2 + r_4\nu^4 - \cdots + (-1)^{p'} r_{2p'} \nu^{2p'}|^{2K}$, where $p' = \lfloor p/2 \rfloor$. Thus, substituting ν^2 with y, we get the equation that fits the inequality from Lemma 3.6. However, the same strategy cannot be applied here since the real part of $(\eta^2+\alpha^2)|c_0(\eta+\alpha i)|^{2K}$ is tangled up with α and η, hence making it impossible to get an equation of the form $y|r(y)|^k$ by simply taking its real part.

Nevertheless, since we have the extra freedom of choosing α, we can select α carefully to make the real part and imaginary part of $|c_0(\eta+\alpha i)|^{2K}$ separable, while keeping the constant $\eta^2+\alpha^2$ large enough. In particular, this can be achieved for

$$\alpha^2 = L\eta - \eta^2.$$

Observe that as long as $\eta \le L$, we have $\alpha \in [0, \sqrt{L\eta - \eta^2}]$, as required in the bound above. It follows that

$$\begin{aligned} &\sup_{\substack{\eta\in[0,L] \\ \alpha\in[0,\sqrt{L\eta-\eta^2}]}} (\eta^2+\alpha^2)|c_0(\eta+\alpha i)|^{2K} \\ &\qquad \ge \sup_{\eta\in[0,L]} L\eta |c_0(\eta+\alpha i)|^{2K} \\ &\qquad = \sup_{\eta\in[0,L]} L\eta |1 + r_1(\eta+\alpha i) + \cdots + r_p(\eta+\alpha i)^p|^{2K}. \end{aligned}$$

Observe that the factor α in the real terms of $(\eta+\alpha i)^j$ has only even order. Therefore, $\mathrm{Re}(c_0(\eta+\alpha i))$ is a polynomial of η and α^2. Since $\alpha^2 = L\eta - \eta^2$, it is actually a polynomial of η exclusively with real-valued coefficients of degree at most p, which we denote as $c_0'(\eta) = 1 + r_1'\eta + r_2'\eta^2 + \cdots + r_p'\eta^p$. Therefore, we get

$$\begin{aligned}\sup_{\eta\in[0,L]} L\eta|c_0(\eta+\alpha i)|^{2K} &\geq \sup_{\eta\in(0,L]} L\eta|\mathrm{Re}(c_0(\eta+\alpha i))|^{2K}\\ &= \sup_{\eta\in(0,L]} L\eta|c_0'(\eta)|^{2K}.\end{aligned}$$

By Lemma 3.6 and $\|\boldsymbol{A}^{-1}\boldsymbol{b}\| = D$, we now have

$$\sup_{F\in\mathcal{F}_{L,D}} \frac{\|F(\boldsymbol{u}_K)\|^2}{\|\boldsymbol{u}^* - \boldsymbol{u}_0\|^2} = \sup_{F\in\mathcal{F}_{L,D}} \frac{\|F(\boldsymbol{u}_K)\|^2}{D^2} \geq \sup_{\eta\in(0,L]} L\eta|c_0'(\eta)|^{2K} \geq \frac{L^2}{80p^2K},$$

and the claimed lower bound follows after rearranging the last inequality. ∎

The implications of Theorem 3.7 are as follows. Among all algorithms that update their iterates as in (3.12) and use constant (independent of the iteration count) step sizes $\beta_{i,k}$ (i.e., algorithms with constant p and $K = \Theta(k)$), GDA is iteration complexity-optimal for minimizing the norm of a cocoercive operator. This means that other standard methods such as the extragradient/mirror-prox [33, 45] method, dual extrapolation [49], or the method of Popov [57], which fall into the same category, cannot attain a convergence rate for minimizing $\|F(\cdot)\|$ that is faster than $1/\sqrt{k}$. Thus, choosing step sizes $\beta_{i,k}$ that depend on the iteration count is essential for achieving the faster $1/k$ rate of Halpern's algorithm (which, as discussed before, corresponds to choosing $K = \Theta(1)$ and $p = \Theta(k)$). Furthermore, this rate is unimprovable for any of the typical iterative methods that take the form from (3.12).

4. Conclusion and future work. We presented a general and unifying potential function-based framework for analyzing the convergence of first-order algorithms under the gradient norm criterion in the settings of convex and min-max optimization. The framework is intuitive in that it provides an interpretation of the mechanism driving the convergence as a trade-off between reducing the norm of the gradient and reducing some notion of an optimality gap.

Many interesting questions for future work remain. In particular, our framework is primarily applicable to Euclidean setups. Thus, it is an intriguing question whether it is possible to generalize it to other normed spaces. We note that beyond the Euclidean setups, the only results with near-optimal convergence for ℓ_p-normed spaces in the setting of convex optimization are those for ℓ_∞ (where an ℓ_∞ variant of gradient descent is optimal) and the very recent results for $p \in [1, 2]$ that are based on a regularization trick [18]. In a different direction, as conjectured in section 2, it appears that fixing either the number of iterations or the accuracy of the problem in advance is crucial for achieving near optimal rates in the case of convex objectives, even in Euclidean setups. Proving such a lower bound would be very interesting, as it would likely require completely new mathematical techniques. Finally, very little is known about the convergence in gradient norm in convex-concave min-max optimization setups, both from the aspect of algorithms and the lower bounds. In particular, we are not aware of any lower bounds outside of the Euclidean setup considered here, while, similar to the case of convex optimization, the only near-optimal algorithm is based on a regularization trick and applies only to $p \in [1, 2]$ [64].

Appendix A. Sequence growth for the optimized gradient method. This section provides a technical lemma used in the proof of Theorem 2.7.

LEMMA A.1. *Let* $\{\beta_{i,k}\}_{i\le k}$, $\{a_k\}_{k\ge 0}$, $\{A_k\}_{k\ge 0}$ *be the sequences of real numbers that for* $k \in \{0,\dots,K\}$ *satisfy* $\beta_{k,k}=1$, $A_k=\sum_{i=0}^{k} a_i$, *and for* $0 \le j < k \le K-1$,

$$\beta_{k,K-1} + \frac{a_{k+1}}{A_K} = \frac{A_{k+1}}{a_{k+1}}, \tag{A.1}$$

$$A_{k+1}\beta_{j,k} = A_k\beta_{j,k-1} + a_{k+1}\Big(\beta_{j,K-1} + \frac{a_{j+1}}{A_K}\Big) + a_{j+1}\Big(\beta_{k,K-1} + \frac{a_{k+1}}{A_K}\Big). \tag{A.2}$$

Then the sequence $\{A_k\}_{k\ge 0}$ *can be chosen as*

$$\begin{cases} A_k = 1 & \text{if } k = K, \\ A_k = A_{k+1}\big[1 + \frac{1}{2}A_{k+1} - \frac{1}{2}\sqrt{A_{k+1}(4+A_{k+1})}\big] & \text{if } 0 \le k \le K-1 \end{cases} \tag{A.3}$$

and $\frac{A_K}{A_0} \ge \frac{(K+2)^2}{4}$.

Proof. First, we show that the sequence $\{A_k\}_{k=0}^{K}$ with $A_K = 1$ that satisfies (A.1) and (A.2) has the following recursive relationship between two successive terms:

$$\frac{1}{A_{k-1}} = \frac{1}{A_k} + \frac{A_k}{a_k}, \tag{A.4}$$

which is equivalent to

$$\frac{A_{k-1}A_k}{a_k} = \frac{a_k}{A_k}. \tag{A.5}$$

Solving for A_{k-1}, this relationship leads to (A.3). We prove the recursive relationship by induction on k. First, for the base case $k = K$, setting $k = K-1$ in (2.19), we have $\beta_{K-1,K-1} = \frac{A_K}{a_K} - \frac{a_K}{A_K}$. Since we have set $A_K = 1$ and $\beta_{K-1,K-1} = 1$, it follows that

$$\frac{a_K}{A_K} = \frac{A_K - a_K}{a_K} = \frac{A_{K-1}}{a_K} = \frac{A_K A_{K-1}}{a_K},$$

which coincides with (A.5).

Now assume that (A.4) (equivalently, (A.5)) holds for $k = K, K-1, \dots, n+1$, and consider $k = n$. Setting $k = n$, $j = n-1$ in (2.20), we have

$$A_{n+1}\beta_{n-1,n} = A_n\beta_{n-1,n-1} + a_{n+1}\frac{A_n}{a_n} + a_n\frac{A_{n+1}}{a_{n+1}}.$$

Hence

$$A_{n+1}\beta_{n-1,n} = A_n + a_{n+1}\frac{A_n}{a_n} + a_n\frac{A_{n+1}}{a_{n+1}}. \tag{A.6}$$

It turns out that we can express $A_{n+1}\beta_{n-1,n}$ using A_k for k ranging from $n+1$ to K. Let $k = \ell$, $\ell = n+1, n+2, \dots, K-1$ and $j = n-1$ in (2.20); then, we get

$$A_\ell\beta_{n-1,l-1} = A_{\ell+1}\beta_{n-1,\ell} - a_{\ell+1}\frac{A_n}{a_n} - a_n\frac{A_{\ell+1}}{a_{\ell+1}}.$$

This is a recursive relation between $A_\ell\beta_{n-1,\ell-1}$ and $A_{\ell+1}\beta_{n-1,\ell}$. Applying this relation

recursively from $\ell = n+1$ to $\ell = K-1$, we get

$$\begin{aligned}
A_{n+1}\beta_{n-1,n} &= A_K\beta_{n-1,K-1} - \frac{A_n}{a_n}(a_{n+2}+\cdots+a_K) - a_n\Big(\frac{A_{n+2}}{a_{n+2}}+\cdots+\frac{A_K}{a_K}\Big)\\
&= A_K\Big(\frac{A_n}{a_n}-\frac{a_n}{A_K}\Big) - \frac{A_n}{a_n}\sum_{\ell=n+1}^{K-1}(A_{\ell+1}-A_\ell) - a_n\sum_{\ell=n+1}^{K-1}\Big(\frac{1}{A_\ell}-\frac{1}{A_{\ell+1}}\Big)\\
&= A_K\Big(\frac{A_n}{a_n}-\frac{a_n}{A_K}\Big) - \frac{A_n}{a_n}(A_K-A_{n+1}) - a_n\Big(\frac{1}{A_{n+1}}-\frac{1}{A_K}\Big)\\
&= \frac{A_nA_{n+1}}{a_n} - \frac{a_n}{A_{n+1}}.
\end{aligned}$$

The second equation is valid due to our inductive hypothesis for $k = n+2, n+3, \ldots, K$. To derive the last equation, we use that $A_K = 1$ by the lemma assumption. Plugging the above equation into (A.6), we get

$$\frac{A_nA_{n+1}}{a_n} = A_n\frac{a_n+a_{n+1}}{a_n} + a_n\Big(\frac{A_{n+1}}{a_{n+1}}+\frac{1}{A_{n+1}}\Big).$$

Using the assumption that $\frac{1}{A_n} = \frac{1}{A_{n+1}} + \frac{A_{n+1}}{a_{n+1}}$, we obtain (A.5) for $k = n$, completing the inductive argument.

The recursive relationship between A_k and A_{k+1} from (2.18) be equivalently written as

$$\frac{1}{A_k} = \frac{1}{4}\Bigg(2+\frac{4}{A_{k+1}}+2\sqrt{1+\frac{4}{A_{k+1}}}\Bigg).$$

Denote $D_n = \frac{1}{A_{K-n}}$ for $n \in \{0,\ldots,K\}$. Then

$$D_n = \frac{1}{2} + D_{n-1} + \sqrt{D_{n-1}+\frac{1}{4}}. \tag{A.7}$$

We prove by induction that

$$D_n \geq \frac{(n+2)^2}{4}. \tag{A.8}$$

As $D_0 = \frac{1}{A_K} = 1$, $D_0 = \frac{(0+2)^2}{4}$ holds by definition. Now suppose also for some $n = j$, $0 \leq j \leq K-1$. Then,

$$\begin{aligned}
D_{j+1} &= \frac{1}{2} + D_j + \sqrt{D_j+\frac{1}{4}}\\
&\geq \frac{1}{2} + \frac{1}{4}(j+2+1-1)^2 + \frac{1}{2}\sqrt{(j+2)^2}\\
&= \frac{1}{2} + \frac{1}{4}(j+3)^2 - \frac{1}{2}(j+3) + \frac{1}{4} + \frac{1}{2}(j+2)\\
&> \frac{1}{4}(j+3)^2.
\end{aligned}$$

Thus, $D_K = \frac{1}{A_0} = \frac{A_K}{A_0} \geq \frac{(K+2)^2}{4}$, as claimed. ∎

Acknowledgments. We thank Adrien Taylor and the anonymous referees for many useful comments and suggestions. Part of this research was done while the second author was an undergraduate student at Shandong University and while he was attending UW-Madison as part of the Visiting International Student Program (VISP).

REFERENCES

[1] Z. Allen-Zhu and L. Orecchia, *Linear coupling: An ultimate unification of gradient and mirror descent*, in Proceedings of the 8th Innovations in Theoretical Computer Science Conference, LIPIcs. Leibniz Int. Proc. Inform. 67, Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2017, 3.

[2] Y. Arjevani, S. Shalev-Shwartz, and O. Shamir, *On lower and upper bounds in smooth and strongly convex optimization*, J. Mach. Learn. Res., 17 (2016), 126.

[3] Y. Arjevani and O. Shamir, *On the iteration complexity of oblivious first-order optimization algorithms*, in Proceedings of the 33rd International Conference on International Conference on Machine Learning, 2016, pp. 908–916.

[4] H. Attouch and F. Alvarez, *The heavy ball with friction dynamical system for convex constrained minimization problems*, in Optimization, Lecture Notes in Econom. and Math. Systems 481, Springer, Berlin, 2000, pp. 25–35.

[5] H. Attouch, J. Bolte, and B. F. Svaiter, *Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods*, Math. Program., 137 (2013), pp. 91–129.

[6] H. Attouch, Z. Chbani, J. Fadili, and H. Riahi, *First-order optimization algorithms via inertial systems with Hessian driven damping*, Math. Program., 193 (2022), pp. 113–155.

[7] H. Attouch, Z. Chbani, and H. Riahi, *Rate of convergence of the Nesterov accelerated gradient method in the subcritical case* $\alpha \leq 3$, ESAIM Control Optim. Calc. Var., 25 (2019), 2.

[8] H. Attouch, X. Goudou, and P. Redont, *The heavy ball with friction method,* I. *The continuous dynamical system: Global exploration of the local minima of a real-valued function by asymptotic analysis of a dissipative dynamical system*, Commun. Contemp. Math., 2 (2000), pp. 1–34.

[9] N. Bansal and A. Gupta, *Potential-function proofs for gradient methods*, Theory Comput., 15 (2019), 4.

[10] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, CMS Books Math./Ouvrages Math. SMC 408, Springer, New York, 2011.

[11] M. Betancourt, M. I. Jordan, and A. C. Wilson, *On Symplectic Optimization*, preprint, https://arxiv.org/abs/1802.03653, 2018.

[12] J. Bolte, A. Daniilidis, O. Ley, and L. Mazet, *Characterizations of Łojasiewicz inequalities: Subgradient flows, talweg, convexity*, Trans. Amer. Math. Soc., 362 (2010), pp. 3319–3363.

[13] H. Brézis and P. L. Lions, *Produits infinis de résolvantes*, Israel J. Math., 29 (1978), pp. 329–345.

[14] S. Bubeck, Y. T. Lee, and M. Singh, *A Geometric Alternative to Nesterov's Accelerated Gradient Descent*, preprint, https://arxiv.org/abs/1506.08187, 2015.

[15] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford, *Lower bounds for finding stationary points* I, Math. Program., 184 (2020), pp. 71–120.

[16] E. De Klerk, F. Glineur, and A. B. Taylor, *Worst-case convergence analysis of inexact gradient and Newton methods through semidefinite programming performance estimation*, SIAM J. Optim., 30 (2020), pp. 2053–2082, https://doi.org/10.1137/19M1281368.

[17] J. Diakonikolas, *Halpern iteration for near-optimal and parameter-free monotone inclusion and strong solutions to variational inequalities*, in Proceedings of the Thirty Third Conference on Learning Theory, Mach. Learn. Res. (PMLR) 125, PMLR, 2020, pp. 1428–1451.

[18] J. Diakonikolas and C. Guzmán, *Complementary Composite Minimization, Small Gradients in General Norms, and Applications to Regression Problems*, preprint, https://arxiv.org/abs/2101.11041, 2021.

[19] J. Diakonikolas and M. I. Jordan, *Generalized momentum-based methods: A Hamiltonian perspective*, SIAM J. Optim., 31 (2021), pp. 915–944, https://doi.org/10.1137/20M1322716.

[20] J. Diakonikolas and L. Orecchia, *Accelerated extra-gradient descent: A novel, accelerated first-order method*, in Proceedings of the 9th Innovations in Theoretical Computer Science Conference, LIPIcs. Leibniz Int. Proc. Inform. 94, Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2018, 23.

[21] J. Diakonikolas and L. Orecchia, *The approximate duality gap technique: A unified theory of first-order methods*, SIAM J. Optim., 29 (2019), pp. 660–689, https://doi.org/10.1137/18M1172314.

[22] Y. Drori and M. Teboulle, *Performance of first-order methods for smooth convex minimization: A novel approach*, Math. Program., 145 (2014), pp. 451–482.

[23] Y. Drori, *The exact information-based complexity of smooth convex minimization*, J. Complexity, 39 (2017), pp. 1–16.

[24] D. Drusvyatskiy, M. Fazel, and S. Roy, *An optimal first order method based on optimal quadratic averaging*, SIAM J. Optim., 28 (2018), pp. 251–271, https://doi.org/10.1137/16M1072528.

[25] N. Golowich, S. Pattathil, C. Daskalakis, and A. Ozdaglar, *Last iterate is slower than averaged iterate in smooth convex-concave saddle point problems*, in Proceedings of COLT'20, 2020, pp. 1758–1784.

[26] G. Gu and J. Yang, *Tight sublinear convergence rate of the proximal point algorithm for maximal monotone inclusion problems*, SIAM J. Optim., 30 (2020), pp. 1905–1921, https://doi.org/10.1137/19M1299049.

[27] B. Halpern, *Fixed points of nonexpanding maps*, Bull. Amer. Math. Soc., 73 (1967), pp. 957–961.

[28] B. Hu and L. Lessard, *Control interpretations for first-order optimization methods*, in Proceedings of the 2017 American Control Conference, 2017, pp. 3114–3119.

[29] M. Ito and M. Fukuda, *Nearly Optimal First-order Methods for Convex Optimization under Gradient Norm Measure: An Adaptive Regularization Approach*, preprint, https://arxiv.org/abs/1912.12004, 2019.

[30] D. Kim, *Accelerated Proximal Point Method and Forward Method for Monotone Inclusions*, preprint, https://arxiv.org/abs/1905.05149, 2019.

[31] D. Kim and J. A. Fessler, *Generalizing the optimized gradient method for smooth convex minimization*, SIAM J. Optim., 28 (2018), pp. 1920–1950, https://doi.org/10.1137/17M112124X.

[32] D. Kim and J. A. Fessler, *Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions*, J. Optim. Theory Appl., 188 (2021), pp. 192–219.

[33] G. Korpelevich, *Extragradient method for finding saddle points and other problems*, Matekon, 13 (1977), pp. 35–49.

[34] M. Krasnosel'skii, *Two remarks on the method of successive approximations*, Uspehi Mat. Nauk (N.S.), 10 (1955), pp. 123–127.

[35] W. Krichene, A. Bayen, and P. L. Bartlett, *Accelerated mirror descent in continuous and discrete time*, in Proceedings of the 28th International Conference on Neural Information Processing Systems, Volume 2, 2015, pp. 2845–2853.

[36] J. Lee, C. Park, and E. K. Ryu, *A geometric structure of acceleration and its role in making gradients small fast*, in Proceedings of the 34th International Conference on Neural Information Processing Systems, 2021.

[37] L. Lessard, B. Recht, and A. Packard, *Analysis and design of optimization algorithms via integral quadratic constraints*, SIAM J. Optim., 26 (2016), pp. 57–95, https://doi.org/10.1137/15M1009597.

[38] F. Lieder, *On the convergence rate of the Halpern-iteration*, Optim. Lett., 15 (2021), pp. 405–418.

[39] H. Lin, J. Mairal, and Z. Harchaoui, *A universal catalyst for first-order optimization*, in Proceedings of the 28th International Conference on Neural Information Processing Systems, Volume 2, 2015, pp. 3384–3392.

[40] Q. Lin and L. Xiao, *An adaptive accelerated proximal gradient method and its homotopy continuation for sparse optimization*, Comput. Optim. Appl., 60 (2015), pp. 633–674.

[41] S. Lojasiewicz, *Une propriété topologique des sous-ensembles analytiques réels*, Les équations aux dérivées partielles, 117 (1963), pp. 87–89.

[42] S. Łojasiewicz, *Ensembles semi-analytiques*, Lectures Notes IHES (Bures-sur-Yvette), 1965.

[43] W. R. Mann, *Mean value methods in iteration*, Proceedings of the Proc. Amer. Math. Soc., 4 (1953), pp. 506–510.

[44] R. D. Monteiro and B. F. Svaiter, *An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods*, SIAM J. Optim., 23 (2013), pp. 1092–1125, https://doi.org/10.1137/110833786.

[45] A. Nemirovski, *Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems*, SIAM J. Optim., 15 (2004), pp. 229–251, https://doi.org/10.1137/S1052623403425629.
[46] A. Nemirovskii and Yudin, *Problem Complexity and Method Efficiency in Optimization*, John Wiley & Sons, New York, 1983.
[47] A. Nemirovsky, *On optimality of Krylov's information when solving linear operator equations*, J. Complexity, 7 (1991), pp. 121–130.
[48] A. Nemirovsky, *Information-based complexity of linear operator equations*, J. Complexity, 8 (1992), pp. 153–175.
[49] Y. Nesterov, *Dual extrapolation and its applications to solving variational inequalities and related problems*, Math. Program., 109 (2007), pp. 319–344.
[50] Y. Nesterov, *How to make the gradients small*, Optima. Mathematical Optimization Society Newsletter, 88 (2012), pp. 10–11.
[51] Y. Nesterov, *Gradient methods for minimizing composite functions*, Math. Program., 140 (2013), pp. 125–161.
[52] Y. Nesterov, *Lectures on Convex Optimization*, Springer Optim. Appl. 137, 2nd ed., Springer, Cham, 2018.
[53] Y. Nesterov, A. Gasnikov, S. Guminov, and P. Dvurechensky, *Primal–dual accelerated gradient methods with small-dimensional relaxation oracle*, Optim. Methods Softw., 35 (2020), pp. 1–38.
[54] Y. E. Nesterov, *A method for solving the convex programming problem with convergence rate $O(1/k^2)$*, Dokl. Akad. Nauk SSSR, 269 (1983), pp. 543–547.
[55] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, New York, 2006.
[56] Y. Ouyang and Y. Xu, *Lower complexity bounds of first-order methods for convex-concave bilinear saddle-point problems*, Math. Program., 185 (2021), pp. 1–35.
[57] L. D. Popov, *A modification of the Arrow-Hurwicz method for search of saddle points*, Math. Notes, 28 (1980), pp. 845–848.
[58] T. Rockafellar, *Monotone operators associated with saddle-functions and minimax problems*, in Proceedings of Symposia in Pure Math, Nonlinear Functional Analysis, Part I, Amer. Math. Soc., Providence, RI, 1970, pp. 241–250.
[59] E. Ryu and W. Yin, *Large-scale Convex Optimization via Monotone Operators*, Cambridge University Press, to appear.
[60] S. Sabach and S. Shtern, *A first order method for solving convex bilevel optimization problems*, SIAM J. Optim., 27 (2017), pp. 640–660, https://doi.org/10.1137/16M105592X.
[61] D. Scieur, V. Roulet, F. Bach, and A. D'Aspremont, *Integration methods and accelerated optimization algorithms*, in Proceedings of Advances in Neural Information Processing Systems 30, 2017.
[62] B. Shi, S. S. Du, M. I. Jordan, and W. J. Su, *Understanding the Acceleration Phenomenon via High-resolution Differential Equations*, preprint, https://arxiv.org/abs/1810.08907, 2018.
[63] C. Song, Y. Jiang, and Y. Ma, *Unified acceleration of high-order algorithms under general Hölder continuity*, SIAM J. Optim., 31 (2021), pp. 1797–1826, https://doi.org/10.1137/19M1290243.
[64] C. Song, Z. Zhou, Y. Zhou, Y. Jiang, and Y. Ma, *Optimistic dual extrapolation for coherent non-monotone variational inequalities*, in Proceedings of the 34th International Conference on Neural Information Processing Systems, 2020, pp. 14303–14314.
[65] W. Su, S. Boyd, and E. J. Candes, *A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights*, J. Mach. Learn. Res., 17 (2016), 153.
[66] A. Taylor and F. Bach, *Stochastic first-order methods: Non-asymptotic and computer-aided analyses via potential functions*, in Proceedings of the International Conference on Learning Theory (COLT), 2019.
[67] A. B. Taylor, J. M. Hendrickx, and F. Glineur, *Exact worst-case performance of first-order methods for composite convex optimization*, SIAM J. Optim., 27 (2017), pp. 1283–1313, https://doi.org/10.1137/16M108104X.
[68] A. B. Taylor, J. M. Hendrickx, and F. Glineur, *Smooth strongly convex interpolation and exact worst-case performance of first-order methods*, Math. Program., 161 (2017), pp. 307–345.
[69] P. Tseng, *On Accelerated Proximal Gradient Methods for Convex-concave Optimization*, 2008.
[70] A. Wibisono, A. C. Wilson, and M. I. Jordan, *A variational perspective on accelerated methods in optimization*, Proc. Natl. Acad. Sci. USA 113 (2016), pp. E7351–E7358.
[71] A. C. Wilson, B. Recht, and M. I. Jordan, *A Lyapunov Analysis of Momentum Methods*

in Optimization, preprint, https://arxiv.org/abs/1611.02635, 2016.

[72] T. Yoon and E. K. Ryu, *Accelerated algorithms for smooth convex-concave minimax problems with $O(1/k^2)$ rate on squared gradient norm*, in Proceedings of ICML'21, 2021.

[73] C. Zalinescu, *Convex Analysis in General Vector Spaces*, World Scientific, River Edge, NJ, 2002.

[74] J. Zhang, A. Mokhtari, S. Sra, and A. Jadbabaie, *Direct Runge-Kutta discretization achieves acceleration*, in Proceedings of Advances in Neural Information Processing Systems 31, 2018.