J. Nonlinear Var. Anal. 10 (2026), No. 3, pp. 597-616
Available online at http://jnva.biemdas.com
https://doi.org/10.23952/jnva.10.2026.3.05

FRANK-WOLFE ALGORITHMS FOR (L_0, L_1)-SMOOTH FUNCTIONS

A.A. VYGUZOV[1,3,4,*], F.S. STONYAKIN[1,2,3], A.V. GASNIKOV[1,3,5]

[1]*Moscow Institute of Physics and Technology, Dolgoprudny, Institutsky lane, 9, Russia*
[2]*Simferopol, Academician Vernadsky Avenue, 4, V. I. Vernadsky Crimean Federal University, Republic of Crimea, Russia*
[3]*Innopolis University, Kazan, Tatarstan, 420500, Russia*
[4]*Adyghe State University, Maikop, Pervomayskaya Str., 208, Russia*
[5]*Steklov Mathematical Institute of Russian Academy of Sciences, Gubkina Str., 8, , Moscow, Russia*

Abstract. We propose a new version of the Frank–Wolfe method, called the (L_0, L_1)-*Frank–Wolfe* algorithm, for optimization problems with (L_0, L_1)-smooth objectives. We demonstrate that this algorithm achieves superior theoretical convergence rates compared to the classical Frank–Wolfe method. In addition, we introduce a novel adaptive procedure, termed the *Adaptive* (L_0, L_1)-*Frank–Wolfe* algorithm, which dynamically adjusts the smoothness parameters to further improve performance and stability. Comprehensive numerical experiments confirm the theoretical results and demonstrate the clear practical advantages of both proposed algorithms over existing Frank–Wolfe variants.

Keywords. Adaptive algorithms; Convex optimization; Frank–Wolfe algorithm; Generalized smoothness; (L_0, L_1)-smoothness.

1. INTRODUCTION

There are many methods for constrained optimization: the projected gradient method ([22]), barrier function methods, and penalty function methods ([2, 18]), among others. However, the Frank–Wolfe method, initially proposed in the seminal work [8] and further generalized in [16], has recently gained significant popularity due to its inexpensive iteration cost compared to the projected gradient method (see [4, Table 1.1] and [6]). Moreover, the Frank–Wolfe method possesses the useful ability to generate sparse solutions, which is beneficial in many applications.

In modern optimization, the Frank–Wolfe method is quite well studied (see [3, 4]). In the seminal work [16], it was proved that the optimal convergence rate of this method is $O(1/k)$ for convex functions; moreover, a series of linear convergence results for the classical Frank–Wolfe method has been established ([3, 4]). However, these results mostly rely on the standard L-smoothness assumption, which is a rather restrictive class of functions. To relax this assumption, attempts have been made to introduce relatively smooth objectives ([24, 27]) for the Frank–Wolfe method, leading to a broader function class—yet many important modern machine learning problems remain uncovered.

*Corresponding author.
E-mail address: al.vyguzov@yandex.ru (A.A. Vyguzov).
Received 4 November 2025; Accepted 20 December 2025; Published online 1 April 2026.

Recently, [30] experimentally observed that modern language modeling (LM) tasks satisfy a property called (L_0, L_1)-smoothness, which generalizes standard L-smoothness. The pioneering work [30] studied the clipped gradient method, and subsequently, many other gradient-based methods have been investigated for this class of functions. Although the (L_0, L_1)-smoothness framework has been extensively studied for gradient-based methods, very few works addressed the Frank-Wolfe (FW) method. A related analysis of a hybrid between the clipped gradient method and the Frank-Wolfe algorithm was proposed in the recent paper [21], but it did not establish linear convergence guarantees. In contrast, our work focuses on the classical (vanilla) Frank-Wolfe method for the convex case and we demonstrate that exploiting the (L_0, L_1)-smooth structure of the objective can lead to significant acceleration benefits.

There exist different step-size rules for the Frank–Wolfe method, but the most popular variants are the decreasing step size $(2/(k+1))$ and the short step size (2.4) (see the overviews in [4] and [3]). The decreasing step size achieves the optimal sublinear convergence rate and does not depend on any function parameters. Therefore, we focus on the short step size (2.4), where the (L_0, L_1)-smoothness parameters can be utilized to achieve improved convergence rates.

Adaptive step-size strategies have recently gained attention in the Frank–Wolfe literature, with several works [1, 12, 20, 27] proposing backtracking schemes that preserve convergence guarantees. However, these methods rely on a single L-smoothness parameter. We extend this framework by introducing a two-parameter adaptive procedure that simultaneously updates L_0 and L_1 based on their relative influence in the composite term $L_0 + L_1\|\nabla f(x_k)\|$. This design enables more responsive yet stable parameter tuning.

The main contributions of this work are as follows. We propose a new variant of the Frank – Wolfe algorithm tailored for (L_0, L_1)-smooth objectives and provide rigorous convergence guarantees. Specifically, we prove linear convergence rates, demonstrating superior performance compared to classical Frank–Wolfe. In the general convex setting, we establish that the algorithm attains the optimal $O(1/k)$ convergence rate, ensuring it is not worse than the standard method. Additionally, we introduce a novel adaptive procedure for independently updating the (L_0, L_1) parameters, which yields improved empirical performance. Numerical experiments confirm that the proposed algorithm consistently outperforms both standard and existing adaptive Frank–Wolfe variants across diverse benchmarks.

2. Preliminaries

In this paper, we consider the minimization problem

$$\min_{x \in Q} f(x),$$

where Q is a convex compact set and f is a convex and (L_0, L_1)-smooth function. Throughout this paper, unless specified otherwise, we use the standard Euclidean norm $\|\cdot\|$ for vectors and the spectral norm $\|\cdot\|$ for matrices.

Definition 2.1. A function f is said to be (L_0, L_1)-*smooth* if, for all x,

$$\|\nabla^2 f(x)\| \leq L_0 + L_1\|\nabla f(x)\|, \tag{2.1}$$

for some constants $L_0, L_1 > 0$.

This notion was introduced in [30], where it was used to explain the superior convergence behavior of the clipped gradient descent method on deep learning problems compared to standard gradient descent (GD). Later, in [31] (see Remark 2.3), this definition was extended from twice differentiable to once differentiable functions. Specifically, a differentiable function f is said to be (L_0, L_1)-smooth if, for all $x, y \in \mathbb{R}^d$ such that $\|x-y\| \leq \frac{1}{L_1}$,

$$\|\nabla f(x) - \nabla f(y)\| \leq (L_0 + L_1 \|\nabla f(y)\|)\|x-y\|. \tag{2.2}$$

This condition generalizes the standard L-smoothness assumption. Indeed, when $L_1 = 0$, inequalities (2.1)–(2.2) reduce to the classical definition of L-smoothness. Hence, the (L_0, L_1)-smoothness condition defines a broader class of functions than the conventional L-smooth case. Motivated by this generalization, we modify the standard shortest-step rule in the Frank–Wolfe method. Let us briefly recall the classical Frank–Wolfe algorithm [8] with the shortest-step rule [16]. The update rule is given by

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2.3}$$

where k is the iteration index, $\alpha_k \in [0,1]$, $d_k = s_k - x_k$, and

$$s_k \in \mathrm{LMO}_Q(\nabla f(x_k)) = \arg\min_{z \in Q} (\nabla f(x_k)^\top z)$$

is the output of the linear minimization oracle (LMO) over Q. In the case of the shortest-step rule, the step size is chosen as

$$\alpha_k := \min\left\{1, \frac{-\nabla f(x_k)^\top d_k}{L\|d_k\|^2}\right\}. \tag{2.4}$$

This observation motivates us to extend the Frank–Wolfe method (2.3) to (L_0, L_1)-smooth objectives by introducing the following step-size rule:

$$\alpha_k := \min\left\{1, \frac{-\nabla f(x_k)^\top d_k}{(L_0 + L_1\|\nabla f(x_k)\|)\,\|d_k\|^2\, e}\right\}. \tag{2.5}$$

It is well known (see, for example, [3, 4, 16]) that the Frank–Wolfe method achieves a linear convergence rate under additional assumptions. Our theoretical analysis shows that the proposed Frank–Wolfe variant, with the step size defined in (2.5) exhibits a superior convergence rate. The acceleration arises from the fact that $L_1 < L$ and $L_0 < L$, where L denotes the standard smoothness parameter. When the solution lies in the interior of a function satisfying the PL condition, our algorithm is significantly faster than the standard method due to the absence of the PL-condition parameter μ in the convergence rate when $L_0 > L_1\|\nabla f(x_k)\|$. A summary of the corresponding convergence rate estimates is provided in Table 1.

Moreover, as mentioned earlier, not all (L_0, L_1)-smooth functions are L-smooth, meaning that our algorithm applies to a broader class of functions. To illustrate this, we list several examples (see [9]):

- $f(x) = \|x\|^n$, where n is a positive integer, with $L_0 = 2n$ and $L_1 = 2n-1$;
- $f(x) = \exp(a^\top x)$, where $L_0 = 0$ and $L_1 = \|a\|$;
- the logistic loss $f(x) = \log(1 + \exp(-a^\top x))$, where $a \in \mathbb{R}^d$, for which $L_0 = 0$ and $L_1 = \|a\|$, while the standard smoothness constant is $L = \|a\|^2$, which is typically much larger than L_1.

In the work [5] (Proposition 1), it was shown that (L_0, L_1)-smoothness implies the next inequality, which we use as an upper bound for our target function.

TABLE 1. Convergence rates of the standard Frank–Wolfe (FW) method with the shortest step size (see, e.g., [4, 16]) and the proposed (L_0,L_1)-FW Algorithm 1 in the convergence rate when under different assumptions. Abbreviations: PL – Polyak-Łojasiewicz condition; C – convex; SC – strongly convex. Here, $B(x,r) \overset{\text{def}}{=} \{z : \|z-x\| \le r\}$, λ denotes the strong convexity constant of the feasible set (see Definition 3.1), μ is the PL-condition constant, T is the number of iterations for which $L_0 \le L_1\|\nabla f(x)\|$, and K is the number of iterations for which $L_0 > L_1\|\nabla f(x)\|$.

Algorithm	**Objective**	**Domain**	**Assumptions**	**Rate**
FW	C; L-smooth	SC	$\|\nabla f(x_k)\| \ge c > 0$	$f(x_k)-f^* \le (f(x_0)-f^*)\max\left\{\frac{1}{2}, 1-\frac{\lambda c}{2L}\right\}^{k-1}$
(L_0,L_1)-FW Algorithm 1	C; (L_0,L_1)-sm.	SC	$\|\nabla f(x_k)\| \ge c > 0$	$f(x_{k+1})-f^* \le (f(x_0)-f^*)\times \max\left\{\frac{1}{2}, 1-\frac{\lambda}{2eL_1}\right\}^{T} \times \max\left\{\frac{1}{2}, 1-\frac{\lambda c}{2eL_0}\right\}^{K}$ (see Th.3.1)
FW	C; PL; L-smooth	C	$B(x^*,r) \subseteq Q$	$f(x_k)-f^* \le (f(x_0)-f^*)\max\left\{\frac{1}{2}, 1-\frac{r^2\mu}{LD^2}\right\}^{k-1}$
(L_0,L_1)-FW Algorithm 1	C; PL; (L_0,L_1)-sm.	C	$B(x^*,r) \subseteq Q$	$f(x_{k+1})-f^* \le (f(x_0)-f^*)\times \max\left\{\frac{1}{2}, 1-\frac{r}{4eL_1D^2}\right\}^{T} \times \max\left\{\frac{1}{2}, 1-\frac{r^2\mu}{2eL_0D^2}\right\}^{K}$ (see Th.3.2)
FW	C; L-smooth	C		$f(x_{k+1})-f^* \le \frac{2LD^2}{k+3}$
(L_0,L_1)-FW Algorithm 1	C; (L_0,L_1)-smooth	C		$f(x^k)-f(x^*) \le \frac{2e\left(L_0+L_1\max_k\|\nabla f(x_k)\|\right)D^2}{k+3}$

Lemma 2.1 (Proposition 1 from [5] Lemma 2.5)**.** *Definition 2.2 holds if and only if, for all* $x,y \in \mathbb{R}^n$,

$$f(y) \le f(x) + \langle \nabla f(x), y-x\rangle + \frac{L_0+L_1\|\nabla f(x)\|}{2}\exp\left(L_1\|x-y\|\right)\|x-y\|^2.$$

We will derive our algorithm from that upper bound. That inequality can be seen as a generalization of the standard quadratic upper bound [18] with $L_1 = 0$.

The Frank–Wolfe gap [3, 4] serves as a standard optimality measure and practical stopping criterion.

Definition 2.2. The Frank–Wolfe gap of a function $f : Q \to \mathbb{R}$ is defined as

$$G(x) = \max_{s\in Q} -\nabla f(x)^\top (s-x).$$

which satisfies the fundamental inequality

$$G(x) \ge \nabla f(x)^\top (x-x^*) \ge f(x)-f^* > 0. \tag{2.6}$$

3. (L_0,L_1)-FW ALGORITHM

The (L_0,L_1)-Frank–Wolfe (FW) algorithm is presented in Algorithm 1.

Algorithm 1 (L_0, L_1)-Frank–Wolfe Algorithm ((L_0, L_1)-FW)

Input: fixed parameters $L_0 > 0$, $L_1 > 0$, and the maximum number of iterations N
for $k = 0, 1, \ldots, N-1$ **do**
  $s_k \leftarrow \mathrm{LMO}(\nabla f(x_k))$
  $d_k \leftarrow s_k - x_k$
  $\alpha_k \leftarrow \min\left\{1, \dfrac{-\nabla f(x_k)^\top d_k}{(L_0 + L_1\|\nabla f(x_k)\|)\,\|d_k\|^2 e}\right\}$
  $x_{k+1} \leftarrow x_k + \alpha_k d_k$
end for
Output: final iterate x_N

In the following sections, we derive convergence estimates for Algorithm 1 under different assumptions. Before doing so, we first establish a descent lemma, formulated for the class of (L_0, L_1)-smooth objective functions.

Lemma 3.1 (Descent Lemma)**.** *Let f be a convex function satisfying the (L_0, L_1)-smoothness condition (Definition 2.2). Then, for Algorithm 1 and every iteration $k \geq 1$, the following inequality holds:*

$$f(x_{k+1}) - f(x_k) \leq \frac{\nabla f(x_k)^\top d_k}{2} \cdot \min\left\{1, \frac{-\nabla f(x_k)^\top d_k}{\big(L_0 + L_1\|\nabla f(x_k)\|\big)\,\|d_k\|^2 e}\right\},$$

where $D = \max_{x,y\in Q}\|x-y\|$ and $d_k = s_k - x_k$ with $s_k \in \mathrm{LMO}(\nabla f(x_k))$.

Proof. Let $a_k = L_0 + L_1\|\nabla f(x_k)\|$. Since f satisfies the (L_0, L_1)-smoothness condition, Lemma 2.1 implies

$$f(x_{k+1}) - f(x_k) \leq \alpha_k \nabla f(x_k)^\top d_k + \frac{a_k}{2} e^{\alpha_k L_1 \|d_k\|} \alpha_k^2 \|d_k\|^2. \tag{3.1}$$

If α_k is chosen so that

$$\|x_{k+1} - x_k\| = \alpha_k \|d_k\| \leq \frac{1}{L_1}, \tag{3.2}$$

then $e^{\alpha_k L_1 \|d_k\|} \leq e$, and inequality (3.1) simplifies to

$$f(x_{k+1}) - f(x_k) \leq \alpha_k \nabla f(x_k)^\top d_k + \frac{a_k e}{2} \alpha_k^2 \|d_k\|^2. \tag{3.3}$$

Minimizing the right-hand side of (3.3) over $\alpha_k \in (0, 1]$ yields

$$\alpha_k^* = \min\left\{1, \frac{-\nabla f(x_k)^\top d_k}{a_k \|d_k\|^2 e}\right\}. \tag{3.4}$$

We verify that α_k^* satisfies (3.2). When $\alpha_k^* < 1$, we have

$$(-\nabla f(x_k)^\top d_k) L_1 \|d_k\| \leq \|\nabla f(x_k)\|\,\|d_k\| L_1 \|d_k\| e \leq a_k \|d_k\|^2 e,$$

which implies

$$\frac{-\nabla f(x_k)^\top d_k}{a_k \|d_k\|^2 e} \leq \frac{1}{L_1 \|d_k\|}.$$

If $\alpha_k^* = 1$, the same inequality follows directly:

$$1 \le \frac{-\nabla f(x_k)^\top d_k}{a_k \|d_k\|^2 e} \le \frac{1}{L_1 \|d_k\|}.$$

Thus, α_k^* satisfies (3.2), and (3.3) holds with $\alpha_k = \alpha_k^*$. Substituting this step size into (3.3) yields

$$\begin{aligned} f(x_{k+1}) - f(x_k) &\le \alpha_k \nabla f(x_k)^\top d_k + \frac{e}{2} a_k \|d_k\|^2 \alpha_k \left(\frac{-\nabla f(x_k)^\top d_k}{a_k \|d_k\|^2 e} \right) \\ &= \frac{\alpha_k}{2} \nabla f(x_k)^\top d_k. \end{aligned} \tag{3.5}$$

Finally, substituting α_k from (3.4) into (3.5) completes the proof. □

3.1. **Linear convergence under strongly convex sets.** We now establish the linear convergence rate of Algorithm 1. For this result, the objective function f is not required to be strongly convex; instead, we assume that the feasible set Q is strongly convex.

Definition 3.1 (Strongly convex sets [16])**.** A set Q is called *strongly convex* if there exists $\lambda > 0$ such that, for any $x, y \in Q$, every point z satisfying $\|z - (x+y)/2\| \le \lambda \|x - y\|^2$ also belongs to Q.

Examples of strongly convex sets include ℓ_2-balls and ellipsoids. Such sets possess several important properties, one of which is the so-called *scaling condition*.

Proposition 3.1 (Scaling condition for strongly convex sets)**.** *Let Q be a strongly convex set with constant λ (Definition 3.1), and let ψ be any nonzero vector. Define $s = \operatorname{argmin}_{y \in Q} \psi^\top y$. Then, for all $x \in Q$, $-\psi^\top (s - x) \ge 2\lambda \|\psi\| \|s - x\|^2$.*

The original statement of this proposition appears in the seminal work of Levitin and Polyak [16] (Theorem 6.1, p. 5). An alternative proof can be found in [4] (Proposition 2.19), where the result was established for a more general class of (α, q)-uniformly convex sets.

Theorem 3.1. *Let f be a convex (L_0, L_1)-smooth function such that $\|\nabla f(x)\| > c > 0$, and let Q be a strongly convex set with constant λ. Then Algorithm 1 satisfies*

$$f(x_{k+1}) - f^* \le (f(x_0) - f^*) \max\left\{\frac{1}{2}, 1 - \frac{\lambda}{2eL_1}\right\}^T \max\left\{\frac{1}{2}, 1 - \frac{\lambda c}{2eL_0}\right\}^K,$$

where T denotes the number of iterations such that $L_0 \le L_1 \|\nabla f(x_k)\|$, and K denotes the number of iterations such that $L_0 > L_1 \|\nabla f(x_k)\|$.

Proof. We distinguish two cases.

Case 1: $\alpha_k < 1$. Substituting the corresponding step size (2.5) into the descent inequality of Lemma 3.1, we obtain

$$\begin{aligned} f(x_k) - f(x_{k+1}) \ge \frac{1}{2e} \cdot \frac{(\nabla f(x_k)^\top d_k)^2}{a_k \|d_k\|^2} &\overset{(3.1)}{\ge} \frac{\lambda}{e} \frac{\|\nabla f(x_k)\|}{a_k} (-\nabla f(x_k)^\top d_k) \overset{(2.6)}{\ge} \\ \frac{\lambda}{e} \frac{\|\nabla f(x_k)\|}{a_k} (f(x_k) - f^*) &= \frac{\lambda}{e} \cdot \frac{\|\nabla f(x_k)\|}{L_0 + L_1 \|\nabla f(x_k)\|} (f(x_k) - f^*). \end{aligned}$$

After straightforward algebraic manipulation, we get

$$f(x_{k+1}) - f^* \le (f(x_k) - f^*)\left(1 - \frac{\lambda}{e} \cdot \frac{\|\nabla f(x_k)\|}{L_0 + L_1\|\nabla f(x_k)\|}\right).$$

We now consider two subcases. If $L_0 \le L_1\|\nabla f(x_k)\|$, then

$$f(x_{k+1}) - f^* \le (f(x_k) - f^*)\left(1 - \frac{\lambda}{2eL_1}\right).$$

Otherwise, when $L_0 > L_1\|\nabla f(x_k)\|$, we have

$$\begin{aligned} f(x_{k+1}) - f^* &\le (f(x_k) - f^*)\left(1 - \frac{\lambda}{e} \cdot \frac{\|\nabla f(x_k)\|}{2L_0}\right) \\ &\le (f(x_k) - f^*)\left(1 - \frac{\lambda c}{2eL_0}\right). \end{aligned}$$

Case 2: $\alpha_k = 1$. In this case, applying Lemma 3.1 with $\alpha_k = 1$ yields

$$f(x_{k+1}) - f(x_k) \le \frac{\nabla f(x_k)^\top d_k}{2} \overset{(2.6)}{\le} -\frac{1}{2}\big(f(x_k) - f^*\big),$$

which implies

$$f(x_{k+1}) - f^* \le \frac{1}{2}\big(f(x_k) - f^*\big).$$

Combining these inequalities gives the claimed linear convergence rate. □

As previously noted, the estimate in Theorem 3.1 improves upon the classical Frank-Wolfe method with the shortest step size, whose rate satisfies

$$f(x_k) - f^* \le (f(x_0) - f^*)\max\left\{\frac{1}{2}, 1 - \frac{\lambda c}{2L}\right\}^{k-1},$$

(see, e.g., [4]), particularly in the regime where $L \gg L_1$ and $L \gg L_0$.

3.2. **Linear convergence under the gradient dominance condition with $x^* \in \mathrm{Int}(Q)$.** We now establish the linear convergence rate of Algorithm 1 under a different set of assumptions. In this setting, we consider objective functions that satisfy the Polyak–Łojasiewicz (PL) inequality.

Definition 3.2 (Polyak–Łojasiewicz condition)**.** A differentiable function f is said to satisfy the *PL condition* if there exists a constant $\mu > 0$ such that

$$\frac{1}{2}\|\nabla f(x)\|^2 \ge \mu\,(f(x) - f^*), \quad \forall x.$$

It is important to note that, in this case, the small parameter μ is absent when $L_0 > L_1\|\nabla f(x_k)\|$, which results in faster convergence of our algorithm compared to the standard Frank–Wolfe method.

The PL condition was introduced by Polyak [23]. This class of functions includes all strongly convex functions. Later, Karimi et al. [10] showed that the PL condition also encompasses several broader families, such as weakly strongly convex functions, functions satisfying the restricted secant inequality, and essentially strongly convex functions. Typical examples include logistic regression loss and least-squares objectives.

In this setting, we relax the assumptions on the feasible set. Specifically, we require only that Q is convex and that the solution x^* lies in its interior, i.e., $x^* \in \mathrm{Int}(Q)$. We now recall a well-known scaling condition that holds under this assumption. Additionally recall that $B(x,r) \overset{\text{def}}{=} \{z : \|z-x\| \leq r\}$ is the ball of radius r around x.

Proposition 3.2 (Scaling condition for convex sets)**.** *Let Q be a convex compact set, and let f be a smooth convex function. If there exists $r > 0$ such that $B(x,r) \subseteq Q$, then, for all $x \in Q$,*

$$-\nabla f(x)^\top (s-x) \geq r\|\nabla f(x)\|.$$

The proof of this result can be found in [4, Proposition 2.16].

We are now ready to prove the linear convergence rate for convex objectives satisfying the PL condition. Notably, this result does not require a lower bound on $\|\nabla f(x_k)\|$, in contrast to Theorem 3.1.

Theorem 3.2. *Assume that f is a convex (L_0, L_1)-smooth function satisfying the PL condition 3.2 with constant $\mu > 0$. If there exists $r > 0$ such that $B(x^*, r) \subseteq Q$ for the solution x^*, then Algorithm 1 satisfies*

$$f(x_{k+1}) - f^* \leq (f(x_0) - f^*) \max\left\{\frac{1}{2}, 1 - \frac{r}{4eL_1D^2}\right\}^T \max\left\{\frac{1}{2}, 1 - \frac{r^2\mu}{2eL_0D^2}\right\}^K,$$

where T denotes the number of iterations for which $L_0 \leq L_1\|\nabla f(x_k)\|$, and K denotes the number of iterations for which $L_0 > L_1\|\nabla f(x_k)\|$.

Proof. As in Theorem 3.1, we consider two cases.

Case 1: Algorithm 1 has $\alpha_k < 1$ at iteration k. Starting from the descent inequality of Lemma 3.1, we have

$$f(x_k) - f(x_{k+1}) \geq \frac{1}{2e} \cdot \frac{(\nabla f(x_k)^\top d_k)^2}{a_k\|d_k\|^2} \overset{(3.2)}{\geq} \frac{r^2\|\nabla f(x_k)\|^2}{2ea_k\|d_k\|^2}.$$

Recalling that $a_k = L_0 + L_1\|\nabla f(x_k)\|$, we distinguish two subcases.

When $L_0 \leq L_1\|\nabla f(x_k)\|$, we obtain

$$f(x_k) - f(x_{k+1}) \geq \frac{r^2\|\nabla f(x_k)\|^2}{4eL_0\|d_k\|^2} \overset{(3.2)}{\geq} \frac{r^2\mu(f(x_k)-f^*)}{2eL_0\|d_k\|^2} \geq \frac{r^2\mu(f(x_k)-f^*)}{2eL_0D^2}.$$

After straightforward algebra, we obtain

$$f(x_{k+1}) - f^* \leq (f(x_k) - f^*)\left(1 - \frac{r^2\mu}{2eL_0D^2}\right). \tag{3.6}$$

In the remaining case, when $L_0 > L_1\|\nabla f(x_k)\|$, we have

$$f(x_k) - f(x_{k+1}) \overset{(3.2)}{\geq} \frac{r(-\nabla f(x_k)^\top d_k)}{4eL_1\|d_k\|^2} \overset{(2.6)}{\geq} \frac{r(f(x_k)-f^*)}{4eL_1\|d_k\|^2} \geq \frac{r(f(x_k)-f^*)}{4eL_1D^2}.$$

Consequently,

$$f(x_{k+1}) - f^* \leq (f(x_k) - f^*)\left(1 - \frac{r}{4eL_1D^2}\right). \tag{3.7}$$

Case 2: $\alpha_k = 1$. Following the same reasoning as in Theorem 3.1, we use the descent inequality of Lemma 3.1 with the appropriate step size:

$$f(x_{k+1}) - f(x_k) \leq \frac{\nabla f(x_k)^\top d_k}{2} \overset{(2.6)}{\leq} -\frac{1}{2}\left(f(x_k) - f^*\right),$$

which implies

$$f(x_{k+1}) - f^* \leq \frac{1}{2}(f(x_k) - f^*).$$

Combining these inequalities yields the claimed linear convergence rate. □

Finally, we compare this result with the classical Frank–Wolfe convergence estimate for L-smooth functions under the same setting:

$$f(x_k) - f^* \leq (f(x_0) - f^*) \max\left\{\frac{1}{2}, 1 - \frac{r^2\mu}{LD^2}\right\}^{k-1}.$$

Once again, we observe an acceleration when $L \gg L_1$ and $L \gg L_0$. However, we emphasize an additional distinction from the classical formulation: in our case, acceleration occurs not only due to the inequalities $L \gg L_1$ and $L \gg L_0$, but also due to the absence of the μ parameter from the PL condition 3.2 in the case $L_0 > L_1\|\nabla f(x_k)\|$. Since μ is typically small, the classical convergence estimate is often slower than the rate established for our proposed algorithm.

4. Adaptation Procedure for the (L_0, L_1)-FW Algorithm

The adaptive variant of Algorithm 1 is presented in Listing 2 (denoted as Adapt (L_0, L_1)-FW).

As mentioned in the introduction, it employs a more flexible adaptation mechanism compared to standard approaches. In our method, the parameters L_0 and L_1 are adjusted individually, in proportion to their respective contributions to the total value $a_k = L_0^{(k)} + L_1^{(k)}\|\nabla f(x_k)\|$. Moreover, both parameters are divided simultaneously but multiplied sequentially, one after another. The corresponding adaptive mechanism is illustrated in lines 7, 8, 17, and 20 of Algorithm 2.

We now show that the adaptive procedure in Algorithm 2 does not deteriorate the convergence rate, and that all convergence theorems preserve their corresponding estimates. Our analysis follows the approach of [19] (Lemma 4), adapted to our setting.

Lemma 4.1. *Assume that f is an (L_0, L_1)-smooth function 2.2 with constants L_0 and L_1, and let n_i denote the number of inequality checks in line 13 of Algorithm 2. Then, for an adaptation factor $\rho > 2$, the following bound holds:*

$$\sum_{i=0}^{N} n_i \leq N\left(1 + \frac{\log(\rho+1)}{\log(\rho-1)}\right) + \frac{1}{\log(\rho-1)}\log\frac{\min\{\rho L_0, L_0^{\max}\} + \min\{\rho L_1, L_1^{\max}\}}{L_0^{(0)} + L_1^{(0)}}$$
$$= O(N). \tag{4.1}$$

Proof. Let $a_k = L_0^{(k)} + L_1^{(k)}\|\nabla f(x_k)\|$, $p_0 = L_0^{(k)}/a_k$, and $p_1 = (L_1^{(k)}\|\nabla f(x_k)\|)/a_k$. From Algorithm 2, during iteration i we have

$$L_0^{(i)} + L_1^{(i)} \geq \frac{(\rho-1)^{n_i-1}}{\rho+1}\left(L_0^{(i-1)} + L_1^{(i-1)}\right).$$

Algorithm 2 Adaptive (L_0, L_1)-Frank-Wolfe Algorithm (Adapt (L_0, L_1)-FW)

Input: initial parameters $L_0^{(0)} > 0$, $L_1^{(0)} > 0$, $L_0^{\max} \gg 0$, $L_1^{\max} \gg 0$, maximum number of iterations N, scaling factor $\rho > 2$
toggle $\leftarrow 0$
for $k = 0, 1, \ldots, N-1$ **do**
    $s_k \leftarrow \mathrm{LMO}(\nabla f(x_k))$
    $d_k \leftarrow s_k - x_k$
    $a_k \leftarrow L_0^{(k)} + L_1^{(k)} \|\nabla f(x_k)\|$
    $L_0^{(k)} \leftarrow L_0^{(k)} / (\rho + L_0^{(k)} / a_k)$
    $L_1^{(k)} \leftarrow L_1^{(k)} / (\rho + (L_1^{(k)} \|\nabla f(x_k)\|) / a_k)$
    while True **do**
        $a_k \leftarrow L_0^{(k)} + L_1^{(k)} \|\nabla f(x_k)\|$
        $\alpha(k) \leftarrow \min\left\{1, \dfrac{-\nabla f(x_k)^\top d_k}{a_k \|d_k\|^2 e}\right\}$
        $x_{k+1} \leftarrow x_k + \alpha_k d_k$
        if $f(x_{k+1}) \le f(x_k) + \alpha(k) \nabla f(x_k)^\top d_k + \frac{a_k e}{2} \alpha(k)^2 \|d_k\|^2$ **then**
            break
        else
            if toggle $= 0$ **then**
                $L_0^{(k)} \leftarrow \min\{L_0^{(k)} \cdot (\rho - L_0^{(k)} / a_k), L_0^{\max}\}$
                toggle $\leftarrow 1$
            else
                $L_1^{(k)} \leftarrow \min\{L_1^{(k)} \cdot (\rho - (L_1^{(k)} \|\nabla f(x_k)\|) / a_k), L_1^{\max}\}$
                toggle $\leftarrow 0$
            end if
        end if
    end while
end for
Output: final point x_N

Taking natural logarithms yields

$$n_i \le 1 + \frac{\log(\rho+1)}{\log(\rho-1)} + \frac{1}{\log(\rho-1)} \log \frac{L_0^{(i)} + L_1^{(i)}}{L_0^{(i-1)} + L_1^{(i-1)}}.$$

Summing over $i = 0, \ldots, N$

$$\begin{aligned}\sum_{i=0}^{N} n_i &\le N + N\frac{\log(\rho+1)}{\log(\rho-1)} + \frac{1}{\log(\rho-1)} \log \frac{L_0^{(N)} + L_1^{(N)}}{L_0^{(0)} + L_1^{(0)}} \\ &\le N\left(1 + \frac{\log(\rho+1)}{\log(\rho-1)}\right) + \frac{1}{\log(\rho-1)} \log \frac{\min\{\rho L_0, L_0^{\max}\} + \min\{\rho L_1, L_1^{\max}\}}{L_0^{(0)} + L_1^{(0)}}\end{aligned}$$

gives the claimed $O(N)$ bound. □

Lemma 4.2 (Descent lemma for adaptive parameters)**.** *Let f be a convex function satisfying (L_0, L_1)-smoothness 2.2. Then, for Algorithm 2 and for each iteration $k \geq 1$, we have*

$$f(x_{k+1}) - f(x_k) \leq \frac{\nabla f(x_k)^\top d_k}{2} \cdot \min\left\{1, \frac{-\nabla f(x_k)^\top d_k}{\left(L_0^{(k)} + L_1^{(k)} \|\nabla f(x_k)\|\right) \|d_k\|^2 e}\right\}.$$

Proof. This follows directly from the fact that, at each iteration, the parameters $L_0^{(k)}$ and $L_1^{(k)}$ are chosen to satisfy inequality (3.1). □

Corollary 4.1. *If f satisfies all assumptions of Theorem 3.1, then Algorithm 2 satisfies*

$$f(x_{k+1}) - f^* \leq (f(x_0) - f^*) \max\left\{\frac{1}{2}, 1 - \frac{\lambda}{2eL_1^{\max}}\right\}^T \max\left\{\frac{1}{2}, 1 - \frac{\lambda c}{2eL_0^{\max}}\right\}^K,$$

where T and K are defined as in Theorem 3.1, and $L_j^{\max} = \max_{i=0,\ldots,k} L_j^{(i)}$.

Corollary 4.2. *If f satisfies all assumptions of Theorem 3.2, then Algorithm 2 satisfies*

$$f(x_{k+1}) - f^* \leq (f(x_0) - f^*) \max\left\{\frac{1}{2}, 1 - \frac{r}{4eL_1^{\max} D^2}\right\}^T \max\left\{\frac{1}{2}, 1 - \frac{r^2\mu}{2eL_0^{\max} D^2}\right\}^K,$$

where T, K, and $L_j^{\max}$ are defined as above.

5. Convergence Rate for the General Convex Case

In this section we prove the suboptimal convergence rate of Algorithm 1 for the general convex case.

Lemma 5.1. *Assume that for a sequence $0 < h_0, \ldots, h_N$ and some $K_{max} > 0$ we have*

$$h_{k+1} \leq h_k \left(1 - \frac{h_k}{K_{max}}\right).$$

Then, for $k \geq 0 \in \mathbb{Z}$,

$$h_{k+1} \leq \frac{K_{max}}{k+3}.$$

Proof. The proof proceeds by induction. For $k = 0$, we have

$$h_1 \leq h_0 \left(1 - \frac{h_0}{K_{\max}}\right) \leq \frac{K_{\max}}{4} \leq \frac{K_{\max}}{3} = \frac{K_{\max}}{0+3},$$

where the inequality follows from the upper bound of the function $f(x) = x - \frac{x^2}{K_{\max}}$ for $x \geq 0$.

Assume the statement holds for some k, and let us prove

$$h_{k+2} \leq \frac{K_{\max}}{k+4}.$$

We consider two cases. If $h_{k+1} \leq \frac{K_{\max}}{k+4}$, the inequality follows immediately. Otherwise, if $h_{k+1} > \frac{K_{\max}}{k+4}$, then

$$h_{k+2} \leq h_{k+1} \frac{k+3}{k+4} \leq \frac{K_{\max}}{k+3} \frac{k+3}{k+4} = \frac{K_{\max}}{k+4},$$

where the second inequality follows from the induction hypothesis. □

Theorem 5.1. *Assume that f is a convex function satisfying the (L_0, L_1)-smoothness condition 2.2. Then, for Algorithm 1, for each iteration $k \geq 1$,*

$$f(x^k) - f(x^*) \leq \frac{2eD^2\left(L_0 + L_1 \max_k \|\nabla f(x_k)\|\right)}{k+3},$$

where $D = \max_{x,y\in Q} \|x-y\|$.

Proof. We use the descent inequality from Lemma 3.1:

$$f(x_{k+1}) - f(x_k) \leq \frac{\nabla f(x_k)^\top d_k}{2} \cdot \min\left\{1, \frac{-\nabla f(x_k)^\top d_k}{(L_0 + L_1\|\nabla f(x_k)\|)\,\|d_k\|^2 e}\right\}. \tag{5.1}$$

We consider two cases.

Case 1: $\alpha(k) = 1$. From (5.1), we have

$$f(x_{k+1}) - f(x_k) \leq \frac{1}{2}\nabla f(x_k)^\top d_k \overset{(2.6)}{\leq} \frac{1}{2}(f^* - f(x_k)),$$

and hence

$$f(x_{k+1}) - f^* \leq \frac{1}{2}(f(x_k) - f^*).$$

Subtracting f^* from both sides of (3.3) with $\alpha(k) = 1$ yields

$$\begin{aligned} f(x_{k+1}) - f^* &\leq -f^* + f(x_k) + \nabla f(x_k)^\top d_k + \frac{a_k}{2}\|d_k\|^2 e \\ &\overset{(2.3)}{\leq} \underbrace{-f^* + f(x_k) + \nabla f(x_k)^\top (x^* - x_k)}_{\leq 0 \text{ (by convexity)}} + \frac{a_k}{2}\|d_k\|^2 e \leq \frac{a_k}{2}\|d_k\|^2 e. \end{aligned}$$

Thus, for $\alpha(k) = 1$, we obtain

$$f(x_{k+1}) - f^* \leq \min\left\{\frac{a_k}{2}\|d_k\|^2 e, \frac{1}{2}(f(x_k) - f^*)\right\}. \tag{5.2}$$

Case 2: $\alpha(k) < 1$. Substituting the step size (3.4) into (5.1) yields

$$f(x_{k+1}) - f(x_k) \leq \frac{1}{2}\frac{(\nabla f(x_k)^\top d_k)^2}{a_k\|d_k\|^2 e}. \tag{5.3}$$

We now establish the convergence rate by induction. For $k = 0$ and $\alpha(0) = 1$, from (5.2) we have

$$f(x_1) - f^* \leq \frac{a_0}{2}\|d_0\|^2 e \leq \frac{2\max\{a_0, a_1\}D^2 e}{4}.$$

For arbitrary $k+1$, the desired inequality follows directly from the induction hypothesis and the right-hand side of (5.2).

If $\alpha(k) < 1$, we slightly reformulate (3.5):

$$f(x_{k+1}) - f^* \leq (f(x_k) - f^*) - \frac{1}{2}\frac{(\nabla f(x_k)^\top d_k)^2}{a_k\|d_k\|^2 e} \overset{(2.6)}{\leq} (f(x_k) - f^*)\left(1 - \frac{\nabla f(x_k)^\top d_k}{2a_k\|d_k\|^2 e}\right).$$

Denoting $h_k = f(x_k) - f^*$ and $K = 2a_k\|d_k\|^2 e$ (note that $h_k \leq K$ by (5.1) and the convexity of f), and applying Lemma 5.1, we obtain

$$f(x_{k+1}) - f^* \leq \frac{2eD^2\left(L_0 + L_1 \max_k \|\nabla f(x_k)\|\right)}{k+3}.$$

□

It is well-known that the $O(1/k)$ rate is optimal for the Frank–Wolfe method [22]. Hence, asymptotically, our estimate matches the standard Frank–Wolfe rate.

6. Numerical Experiments

In this section, we present numerical experiments illustrating the performance of the proposed Algorithm 1 ((L_0, L_1)-FW) and its adaptive variant, Algorithm 2 (Adapt (L_0, L_1)-FW). The aim of these experiments is to demonstrate the advantages of the proposed step-size selection rule compared with classical and previously known adaptive Frank–Wolfe variants.

6.1. Objective function.

We consider the optimization of the logistic regression objective

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp(-y_i (Ax)_i)\right),$$

where $x \in \mathbb{R}^d$. This function is smooth in the classical sense and also (L_0, L_1)-smooth. The classical smoothness constant is $L = \max_i \|A_i\|^2$, and the (L_0, L_1)-smoothness constants are set to $L_0 = 0$ and $L_1 = \max_i \|A_i\|$, typically with $\|A_i\| \gg 1$.

6.2. Data generation.

The matrix A and vector y were generated using standard normal distributions:

$$A \sim \mathcal{N}(0, 1).$$

Each column of A was multiplied by a scalar 2^j, where j is the column index. The label vector y was then generated as

$$y = \operatorname{sign}(A^\top x_{\text{sol}} + \text{noise}),$$

where x_{sol} is a randomly generated solution vector, and sign returns the sign of each element.

In all cases, the data are uncorrelated and slightly noisy. We considered three geometric settings for the data distribution:

(1) Random points uniformly distributed on the ℓ_2-ball of radius 25, with increasing number of features and dimension.
(2) Random points uniformly distributed on the 1-simplex, with increasing dimension.
(3) Random points uniformly distributed on the ℓ_∞-ball, with increasing dimension.

6.3. Compared algorithms.

The following algorithms were compared:

- **Classic FW**: the standard Frank–Wolfe algorithm using the shortest-step rule

$$\alpha_k = \min\left\{\frac{\nabla f(x_k)^\top (s_k - x_k)}{L\|s_k - x_k\|^2}, 1\right\},$$

  where L is the Lipschitz constant of the gradient and s_k is the solution of the linear minimization oracle (LMO).
- **Adaptive Classic FW**: identical to the classical Frank–Wolfe algorithm, but with the smoothness constant L updated adaptively at each iteration, as analyzed in [1].
- **(L_0, L_1)-FW**: proposed algorithm (Algorithm 1), with fixed parameters (L_0, L_1) set in advance, e.g., known for the logistic regression objective.

- **Adapt (L_0, L_1)-FW**: proposed algorithm (Algorithm 2), which employs an adaptive step-size rule based on the constants (L_0, L_1), with sequential updates for each parameter.

6.4. **Results and discussion.** The experimental results demonstrate that the additional adaptive correction in the proposed Adapt (L_0, L_1)-FW algorithm consistently accelerates convergence across all test cases. This algorithm achieved the best performance in all experiments. Similarly, the (L_0, L_1)-FW algorithm also significantly outperforms the standard Frank–Wolfe algorithm.

The improvement provided by the adaptive procedure stems from the finer tuning of the step size: the adaptive update in Adapt (L_0, L_1)-FW increases the parameters more smoothly and conservatively, avoiding abrupt changes that could hinder convergence. Additionally, alternating the parameter updates enhances the convergence speed, as the aggregate value

$$a_k = L_0 + L_1 \|\nabla f(x_k)\|$$

increases more gradually and stabilizes earlier than in more aggressive adaptive schemes.

7. Conclusion

In this work, we introduced two new versions of the Frank–Wolfe method for (L_0, L_1)-smooth objective functions: the (L_0, L_1)-*FW* algorithm and its adaptive counterpart, the *Adaptive* (L_0, L_1)-*FW* algorithm. For both methods, we established convergence guarantees under multiple settings, including general convex objectives, convex objectives over strongly convex feasible sets, and convex objectives satisfying the Polyak–Łojasiewicz (PL) condition with the solution in the interior of the feasible set.

The numerical experiments confirm the theoretical results and show that the proposed algorithms outperform the classical Frank–Wolfe method and existing adaptive variants. In particular, the adaptive version exhibits a clear advantage due to its capability to dynamically adjust the parameters L_0 and L_1 based on the local problem geometry.

Our adaptation strategy differs from previous approaches that adjust only a single Lipschitz-like constant. Instead, we adapt both parameters simultaneously, accounting for their relative contributions in the composite smoothness model, which renders the procedure more flexible and responsive.

Despite these encouraging results, several open questions remain. For instance, how large can the adaptive parameters become relative to the true (L_0, L_1) values? And what are the optimal initial choices of L_0 and L_1 when these parameters are not known a priori?

Acknowledgements
This work was supported by the Ministry of Science and Higher Education of the Russian Federation within state assignment no. 075-03-2024-074 under the project "Study of Asymptotic Characteristics of Fluctuations of Differential Equations and Systems, and Optimization Methods."

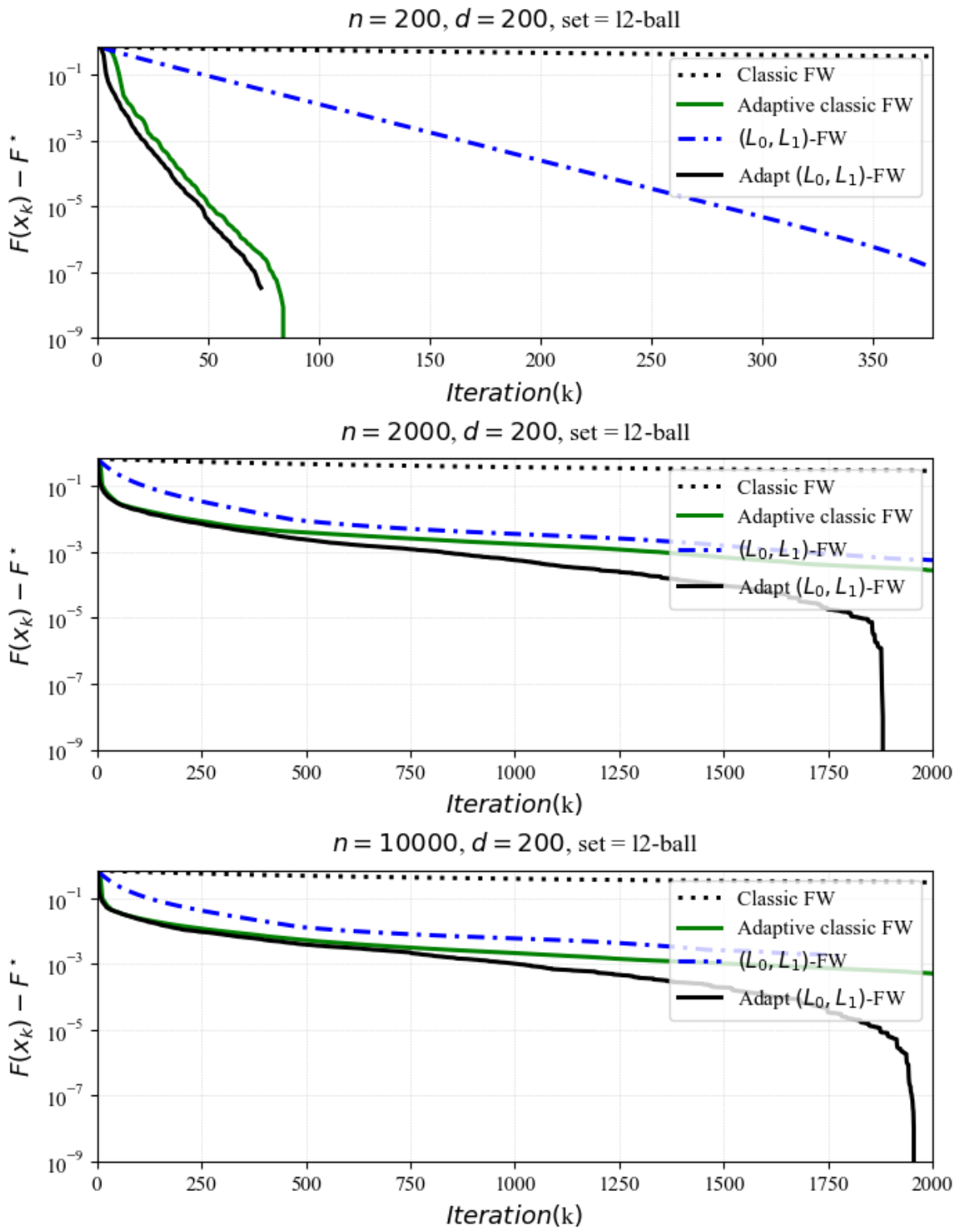

FIGURE 1. ℓ_2-ball set: increasing number of data points (log scale on the Y-axis).

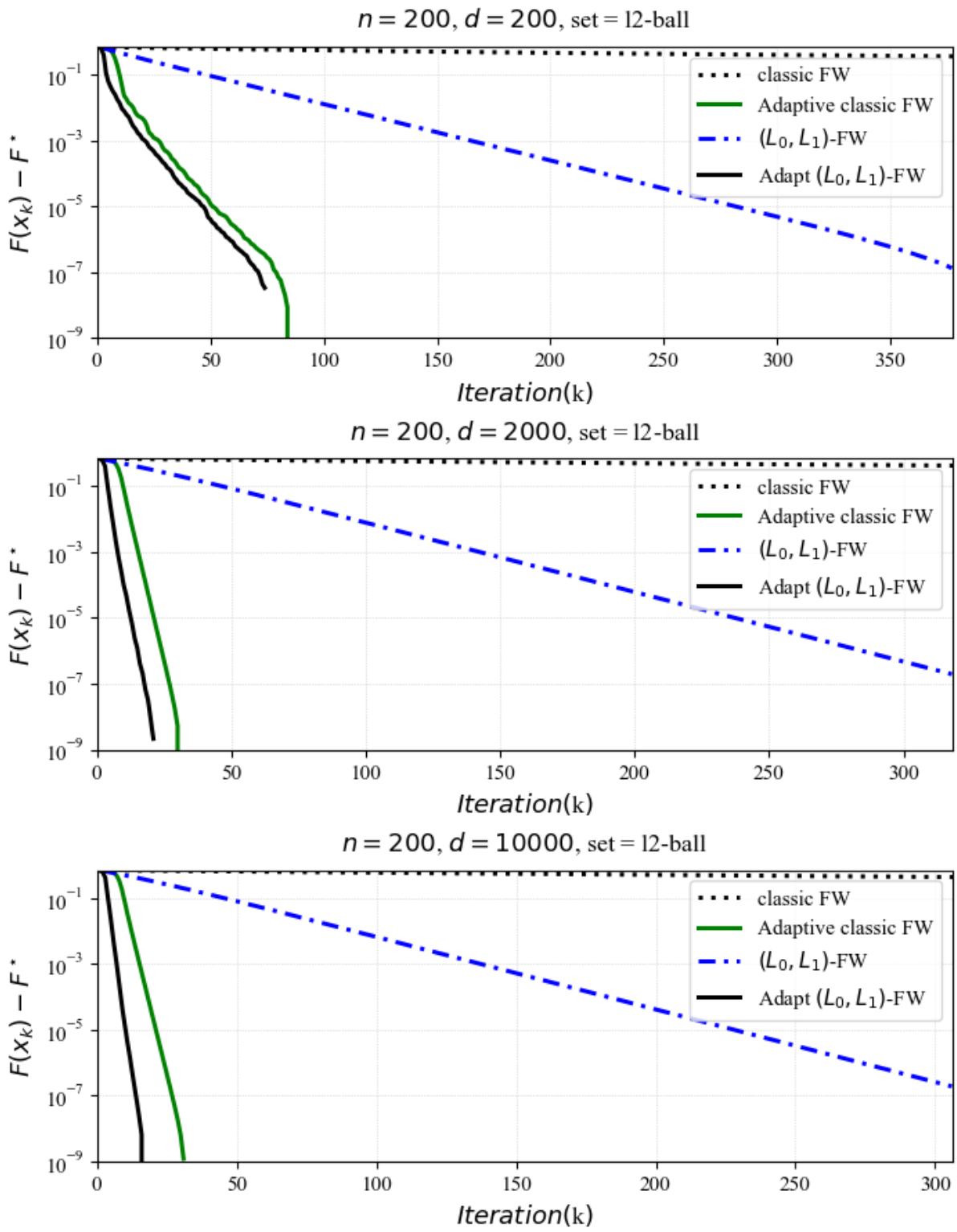

FIGURE 2. ℓ_2-ball set: increasing dimension (log scale on the Y-axis).

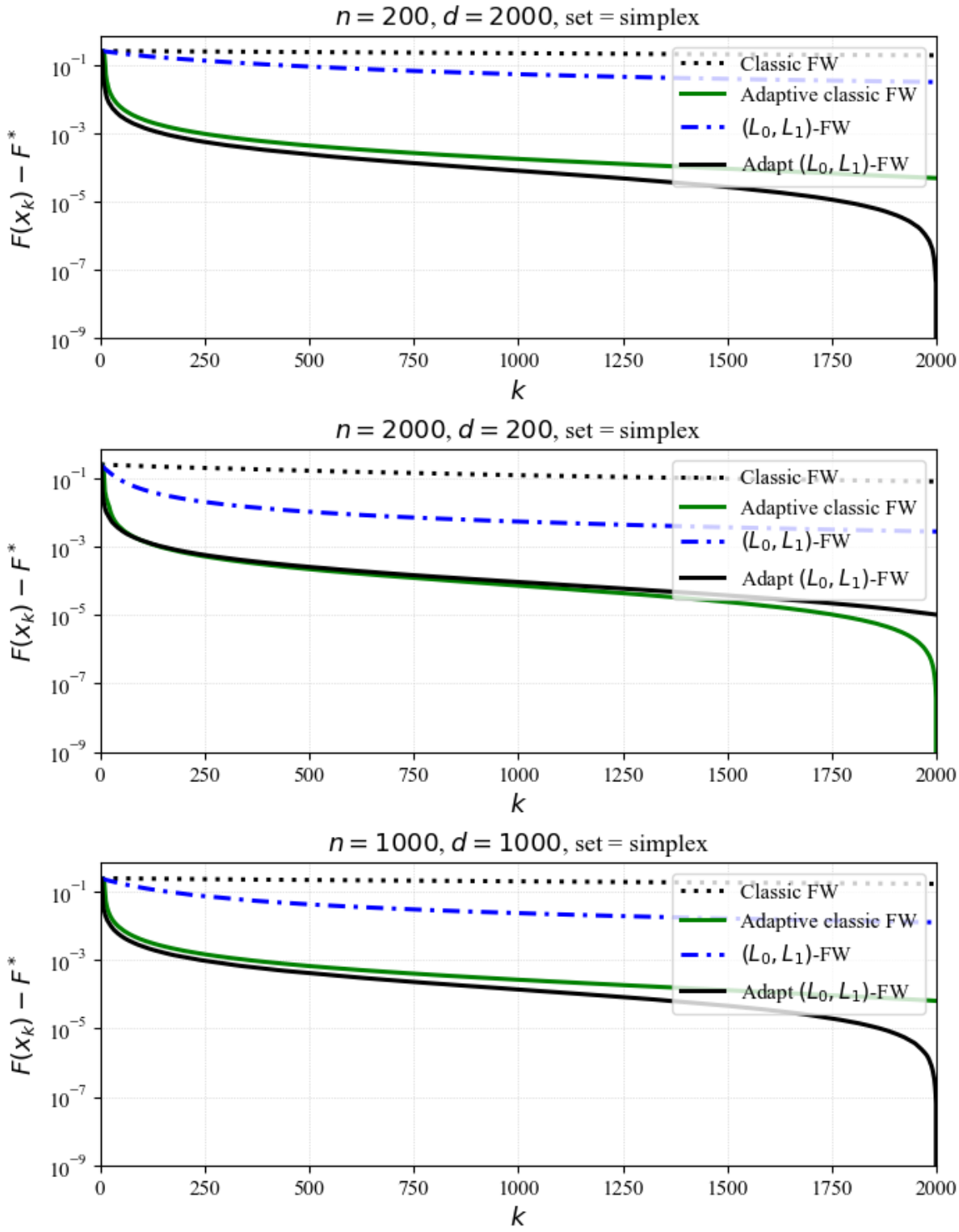

FIGURE 3. 1-simplex set: increasing dimension (log scale on the Y-axis).

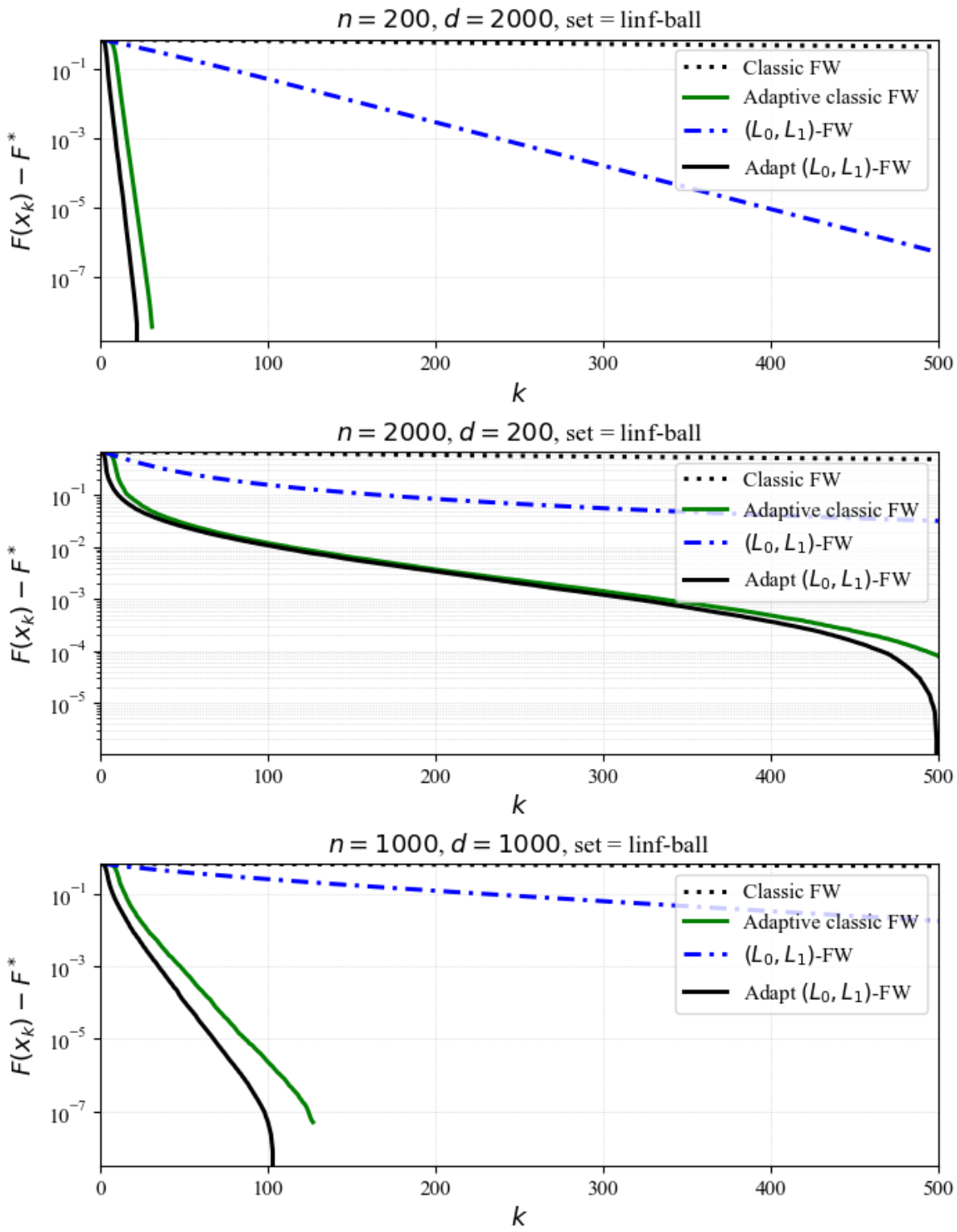

FIGURE 4. ℓ_∞-ball set: increasing dimension (log scale on the Y-axis).

REFERENCES

[1] G. V. Aivazian, F. S. Stonyakin, D. A. Pasechnyk, M. S. Alkousa, A. M. Raigorodsky, I. V. Baran, Adaptive variant of the Frank–Wolfe algorithm for convex optimization problems, Program. Comput. Softw. 49 (2023), 493–504.

[2] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, Cambridge, 2004.

[3] I. M. Bomze, F. Rinaldi, D. Zeffiro, Frank–Wolfe and friends: a journey into projection-free first-order optimization methods, 4OR 19 (2021), 313–345.

[4] G. Braun, A. Carderera, C. W. Combettes, H. Hassani, A. Karbasi, A. Mokhtari, S. Pokutta, Conditional gradient methods, arXiv preprint arXiv:2211.14103, 2022.

[5] Z. Chen, Y. Zhou, Y. Liang, Z. Lu, Generalized-smooth nonconvex optimization is as efficient as smooth nonconvex optimization, In: International Conference on Machine Learning, PMLR, pp. 5396–5427, 2023.

[6] C. W. Combettes, S. Pokutta, Complexity of linear minimization and projection on some sets, Oper. Res. Lett. 49 (2021), 565–571.

[7] M. Crawshaw, M. Liu, F. Orabona, W. Zhang, Z. Zhuang, Robustness to unbounded smoothness of generalized signSGD, Advances in Neural Information Processing Systems, 35 (2022), 9955–9968.

[8] M. Frank, P. Wolfe, An algorithm for quadratic programming, Naval Research Logistics Quarterly, 3 (1956), 95–110.

[9] E. Gorbunov, N. Tupitsa, S. Choudhury, A. Aliev, P. Richtárik, S. Horváth, M. Takáč, Methods for convex (L_0, L_1)-smooth optimization: Clipping, acceleration, and adaptivity, arXiv preprint arXiv:2409.14989, 2024.

[10] H. Karimi, J. Nutini, M. Schmidt, Linear convergence of gradient and proximal-gradient methods under the Polyak–Łojasiewicz condition, In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 795–811, Springer, 2016.

[11] T. Kerdreux, A. d'Aspremont, S. Pokutta, Projection-free optimization on uniformly convex sets, In: International Conference on Artificial Intelligence and Statistics, PMLR, pp. 19–27, 2021.

[12] A. Khademi, A. Silveti-Falls, Adaptive conditional gradient descent, arXiv preprint arXiv:2510.11440, 2025.

[13] A. Koloskova, H. Hendrikx, S. U. Stich, Revisiting gradient clipping: stochastic bias and tight convergence guarantees, In: International Conference on Machine Learning, PMLR, 17343–17363, 2023.

[14] L. Lamport, LaTeX: A Document Preparation System, Addison-Wesley, 1994.

[15] A. Lobanov, A. Gasnikov, E. Gorbunov, M. Takáč, Linear convergence rate in convex setup is possible! Gradient descent method variants under (L_0, L_1)-smoothness, arXiv preprint arXiv:2412.17050, 2024.

[16] E. S. Levitin and B. T. Polyak, Constrained minimization methods, USSR Comput. Math. Math. Phys. 6 (1966), 1–50.

[17] H. Li, J. Qian, Y. Tian, A. Rakhlin, A. Jadbabaie, Convex and non-convex optimization under generalized smoothness, Advances in Neural Information Processing Systems, 36 (2023), 40238–40271.

[18] Y. Nesterov, Lectures on Convex Optimization, Springer, 2018.

[19] Y. Nesterov, Gradient methods for minimizing composite functions, Math. Program. 140 (2013), 125–161.

[20] F. Pedregosa, G. Negiar, A. Askari, M. Jaggi, Linearly convergent Frank–Wolfe with backtracking line-search, In: International Conference on Artificial Intelligence and Statistics, PMLR, pp. 1-10, 2020.

[21] T. Pethick, W. Xie, M. Erdogan, K. Antonakopoulos, T. Silveti-Falls, V. Cevher, Generalized gradient norm clipping & non-Euclidean (L_0, L_1)-smoothness, arXiv preprint arXiv:2506.01913, 2025.

[22] B. T. Polyak, Vvedenie v Optimizatsiyu [Introduction to Optimization], Nauka, Moscow, 1983.

[23] B. T. Polyak, Gradient methods for minimizing functionals, Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, 3 (1963), 643–653.

[24] S. Takahashi, S. Pokutta, A. Takeda, Fast Frank–Wolfe algorithms with adaptive Bregman step-size for weakly convex functions, arXiv preprint arXiv:2504.04330, 2025.

[25] Y. Takezawa, H. Bao, R. Sato, K. Niwa, M. Yamada, Polyak meets parameter-free clipped gradient descent, CoRR, 2024.

[26] D. Vankov, A. Rodomanov, A. Nedich, L. Sankar, S. U. Stich, Optimizing (L_0, L_1)-smooth functions by gradient methods, arXiv preprint arXiv:2410.10800, 2024.

[27] A. A. Vyguzov, F. S. Stonyakin, An adaptive variant of the Frank–Wolfe method for relative smooth convex optimization problems, Comput. Math. Math. Phys. 65 (2025), 591–602.

[28] B. Wang, H. Zhang, Z. Ma, W. Chen, Convergence of AdaGrad for non-convex objectives: Simple proofs and relaxed assumptions, In: The Thirty Sixth Annual Conference on Learning Theory, PMLR, 161–190, 2023.
[29] B. Wang, Y. Zhang, H. Zhang, Q. Meng, R. Sun, Z. Ma, T.-Y. Liu, Z.-Q. Luo, W. Chen, Provable adaptivity of Adam under non-uniform smoothness, In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 2960–2969, 2024.
[30] J. Zhang, T. He, S. Sra, A. Jadbabaie, Why gradient clipping accelerates training: A theoretical justification for adaptivity, arXiv preprint arXiv:1905.11881, 2019.
[31] B. Zhang, J. Jin, C. Fang, L. Wang, Improved analysis of clipping algorithms for non-convex optimization, Advances in Neural Information Processing Systems, 33 (2020), 15511–15521.
[32] S.-Y. Zhao, Y.-P. Xie, W.-J. Li, On the convergence and improvement of stochastic normalized gradient descent, Sci. Chin. Info. Sci. 64 (2021), 132103.