Linearization Algorithms for Fully Composite Optimization

Maria-Luiza Vladarean MARIA-LUIZA.VLADAREAN@EPFL.CH
École Polytechnique Fédérale de Lausanne

Nikita Doikov NIKITA.DOIKOV@EPFL.CH
École Polytechnique Fédérale de Lausanne

Martin Jaggi MARTIN.JAGGI@EPFL.CH
École Polytechnique Fédérale de Lausanne

Nicolas Flammarion NICOLAS.FLAMMARION@EPFL.CH
École Polytechnique Fédérale de Lausanne

Editors: Gergely Neu and Lorenzo Rosasco

Abstract

This paper studies first-order algorithms for solving fully composite optimization problems over convex and compact sets. We leverage the structure of the objective by handling its differentiable and non-differentiable components separately, linearizing only the smooth parts. This provides us with new generalizations of the classical Frank-Wolfe method and the Conditional Gradient Sliding algorithm, that cater to a subclass of non-differentiable problems. Our algorithms rely on a stronger version of the linear minimization oracle, which can be efficiently implemented in several practical applications. We provide the basic version of our method with an affine-invariant analysis and prove global convergence rates for both convex and non-convex objectives. Furthermore, in the convex case, we propose an accelerated method with correspondingly improved complexity. Finally, we provide illustrative experiments to support our theoretical results.

Keywords: convex optimization, composite problems, Frank-Wolfe algorithm, acceleration

1. Introduction

In this paper we consider fully composite optimization problems of the form

$$\min_{\mathbf{x}\in\mathcal{X}} \Big[\varphi(\mathbf{x}) \overset{\text{def}}{=} F(\mathbf{f}(\mathbf{x}), \mathbf{x}) \Big], \tag{1}$$

where $\mathcal{X}$ is a convex and compact set, $F : \mathbb{R}^n \times \mathcal{X} \to \mathbb{R}$ is a simple but possibly *non-differentiable* convex function and $\mathbf{f} : \mathcal{X} \to \mathbb{R}^n$ is a smooth mapping, which is the *main source of computational burden*.

Problems of this type cover and generalize many classical use-cases of composite optimization and are often encountered in applications. In this work, we develop efficient algorithms for solving (1) by leveraging the *structure of the objective* and using the *linearization principle*. Our method generalizes the well-known Frank-Wolfe algorithm (Frank and Wolfe, 1956) and ensures asymptotically faster convergence rates compared to methods treating φ in a black-box fashion.

A classical algorithm for solving smooth optimization problems is the Gradient Descent method (GD), proposed by Cauchy in 1847 (see historical note by Lemaréchal (2012)). It rests on the idea of linearizing the function around the current iterate, taking a step in the negative gradient direction and projecting the result onto the feasible set $\mathcal{X}$ for $k \geq 0$:

$$\mathbf{y}_{k+1} \quad = \quad \pi_{\mathcal{X}}\big(\mathbf{y}_k - \alpha_k \nabla \varphi(\mathbf{y}_k)\big), \qquad \alpha_k > 0, \tag{2}$$

where $\pi_{\mathcal{X}}$ is the projection operator onto $\mathcal{X}$. Surprisingly, the same kind of iterations can minimize *general* non-smooth convex functions by substituting $\nabla \varphi(\mathbf{y}_k)$ with any *subgradient* in the subdifferential $\partial \varphi(\mathbf{y}_k)$. The resulting Subgradient method was proposed by Shor et al. (1985).

Another notable example for smooth optimization over a convex and *bounded* constraint set $\mathcal{X}$ is the Frank-Wolfe (FW) method (Frank and Wolfe, 1956). Again, a linearization of the objective around the current iterate is used to query the so-called *linear minimization oracle* (LMO) associated with $\mathcal{X}$, for every $k \geq 0$:

$$\mathbf{y}_{k+1} \quad \in \quad \operatorname*{Argmin}_{\mathbf{x}} \big\{ \langle \nabla \varphi(\mathbf{y}_k), \mathbf{x} \rangle \ : \ \mathbf{x} \in \mathbf{y}_k + \gamma_k(\mathcal{X} - \mathbf{y}_k) \big\}, \qquad \gamma_k \in (0, 1]. \tag{3}$$

Steps of type (3) can be significantly cheaper than those involving projections (2) for a few notable domains such as nuclear norm balls and spectrahedrons (Combettes and Pokutta, 2021), making FW the algorithm of choice in such scenarios. Moreover, the solutions found by FW methods can benefit from additional properties such as sparsity (Jaggi, 2013). These desirable features make FW methods suitable for large scale optimization, a fact which led to an increased interest in recent years (we point the reader to the monograph of Braun et al. (2022) for a detailed presentation). Unfortunately, the vanilla FW algorithm does not extend to non-differentiable problems in the same straightforward manner as GD – a counterexample is given by Nesterov (2018a). The question of developing non-smooth versions of the FW algorithm therefore remains open, and is the main focus of this article.

Finally, we touch on the issue of convergence rates – a principal means of theoretically characterizing optimization algorithms. The classical monograph of Nemirovski and Yudin (1983) establishes that the $\mathcal{O}(1/\sqrt{k})$ rate of the Subgradient method is optimal for *general non-differentiable* convex problems, while the $\mathcal{O}(1/k)$ rate of its counterpart GD is far from the lower bound of $\Omega(1/k^2)$ for $L-$smooth convex functions. Similar results are established by Lan (2013) for LMO-based algorithms, although in this case the $\mathcal{O}(1/k)$ rate is matched by a lower bound for smooth convex minimization. This relatively slow convergence of FW algorithms is a result of their *affine-invariant* oracle, which is independent on the choice of norm. In light of these lower bounds, one can only hope to improve convergence rates by imposing additional structure on the problem to be solved.

The present work leverages this observation and studies a subclass of (possibly) non-smooth and non-convex problems with the *specific structure* of (1). Our methods require only linearizations of the differentiable component $\mathbf{f}$, while the non-differentiable function F is kept as a part of the subproblem solved within oracle calls. We show that this approach is a viable way of generalizing FW methods to address problem (1), with the possibility of acceleration. Our contributions can be summarized as follows.

- We propose a basic method for problem (1), which is *affine-invariant* and equipped with accuracy certificates. We prove the global convergence rate of $\mathcal{O}(1/k)$ in the convex setting, and of $\tilde{\mathcal{O}}(1/\sqrt{k})$ in the non-convex case.
- We propose an accelerated method with inexact proximal steps which attains a convergence rate of $\mathcal{O}(1/k^2)$ for convex problems. Our algorithm achieves the optimal $\mathcal{O}\left(\varepsilon^{-1/2}\right)$ oracle complexity for smooth convex problems in terms of the number of computations of $\nabla \mathbf{f}$.
- We provide proof-of-concept numerical experiments, that demonstrate the efficiency of our approach for solving composite problems.

Related Work. The present work lies at the intersection of two broad lines of study: general methods for composite optimization and FW algorithms. The former category encompasses many approaches that single out non-differentiable components in the objective's structure, and leverage this knowledge in the design of efficient optimization algorithms. This approach originated in the works of Burke (1985, 1987); Nesterov (1989); Nemirovski (1995); Pennanen (1999); Boţ et al. (2007, 2008). A popular class of *additive* composite optimization problems was proposed by Beck and Teboulle (2009); Nesterov (2013) and the modern algorithms for general composite formulations were developed by Cui et al. (2018); Drusvyatskiy and Lewis (2018); Drusvyatskiy and Paquette (2019); Bolte et al. (2020); Burke et al. (2021); Doikov and Nesterov (2022).

The primitive on which most of the aforementioned methods rely is a *proximal-type* step – a generalization of (2). Depending on the geometry of the set $\mathcal{X}$, such steps may pose a significant computational burden. Doikov and Nesterov (2022) propose an alternative *contracting-type* method for *fully composite* problems, which generalizes the vanilla FW algorithm. Their method relies on a simpler primitive built on the linearization principle, which can be much cheaper in practice. We study the same problem structure as Doikov and Nesterov (2022) and devise methods with several advantages over the aforementioned approach, including an *affine-invariant analysis*, *accuracy certificates*, convergence guarantees for non-convex problems and, in the convex case, an accelerated convergence. Moreover, we decouple stepsize selection from the computational primitive, to enable efficient line search procedures.

Our methods are also intimately related to FW algorithms, which they generalize. For smooth and convex problems, vanilla FW converges at the cost of $\mathcal{O}\left(\varepsilon^{-1}\right)$ LMO and *first order oracle* (FO) calls in terms the Frank-Wolfe gap – an accuracy measure bounding functional suboptimality (Jaggi, 2013). For smooth non-convex problems, a gap value of at most ε is attained after $\mathcal{O}\left(\varepsilon^{-2}\right)$ LMO and FO calls (Lacoste-Julien, 2016). Due to the relatively slow convergence of LMO-based methods, recent efforts have gone into devising variants with improved guarantees. The number of FO calls was reduced to the lower bound for smooth convex optimization by Lan and Zhou (2016), local acceleration was achieved following a burn-in phase by Diakonikolas et al. (2020); Carderera et al. (2021); Chen and Sun (2022), and empirical performance was enhanced by adjusting the update direction with gradient information by Combettes and Pokutta (2020). Of the aforementioned works, closest to ours is the Conditional Gradient Sliding (CGS) algorithm proposed by Lan and Zhou (2016) and further studied by Yurtsever et al. (2019); Qu et al. (2018). CGS uses the acceleration framework of Nesterov (1983) and solves the projection subproblem inexactly via the FW method, achieving the optimal complexity of $\mathcal{O}\left(\varepsilon^{-1/2}\right)$ FO calls for smooth convex problems. We rely on a similar scheme for improving FO complexity in the convex case.

In the context of generic non-smooth convex objectives, the FW algorithm was studied by Lan (2013), who proposes a smoothing-based approach matching the lower bound of $\Omega(\varepsilon^{-2})$ LMO calls. The method however requires $\mathcal{O}\left(\varepsilon^{-4}\right)$ FO calls, a complexity which is later improved to $\mathcal{O}\left(\varepsilon^{-2}\right)$ by Garber and Hazan (2016) through a modified LMO for polytopes, by Ravi et al. (2019) with a (differently) modified LMO, and finally by Thekumparampil et al. (2020) through a combination of smoothing and the CGS algorithm. Our algorithm, instead, leverages the structure of problem (1) and a modified LMO to achieve improved rates, with the added benefit of an affine invariant method and analysis. We also mention FW methods for additive composite optimization (Argyriou et al., 2014; Yurtsever et al., 2018, 2019; Zhao and Freund, 2022), with the former three relying on proximal steps and the latter assuming a very restricted class of objectives.

Reference	φ subclass	Use structure?	# FO	# PO/LMO	Observations
Shor et al. (1985)	cvx, L-cont	no	$\mathcal{O}\left(\varepsilon^{-2}\right)$ (1)	$\mathcal{O}\left(\varepsilon^{-2}\right)$ (1)	projection
Thekumparampil et al. (2020)	cvx, L-cont	no	$\mathcal{O}\left(\varepsilon^{-2}\right)$ (1)	$\mathcal{O}\left(\varepsilon^{-2}\right)$ (1)	smoothing, vanilla LMO
Doikov and Nesterov (2022)	cvx, fully-comp	yes	$\mathcal{O}\left(\varepsilon^{-1}\right)$ (1)	$\mathcal{O}\left(\varepsilon^{-1}\right)$ (1)	modif. LMO
(this work) Alg. 2	cvx, fully-comp.	yes	$\mathcal{O}\left(\varepsilon^{-1/2}\right)$ (1)	$\mathcal{O}\left(\varepsilon^{-1}\right)$ (1)	modif. LMO
De Oliveira (2023)	non-cvx, upper-$C^{1,\alpha}$	no	$\mathcal{O}\left(\varepsilon^{-2}\right)$ (2)	$\mathcal{O}\left(\varepsilon^{-2}\right)$ (2)	vanilla LMO
Kreimeier et al. (2023)	non-cvx, abs-smooth	no	$\mathcal{O}\left(\varepsilon^{-2}\right)$ (3)	$\mathcal{O}\left(\varepsilon^{-2}\right)$ (3)	modif. LMO
Drusvyatskiy and Paquette (2019)	non-cvx, comp	yes	$\mathcal{O}\left(\varepsilon^{-2}\right)$ (4)	$\mathcal{O}\left(\varepsilon^{-2}\right)$ (4)	prox. steps
(this work) Alg. 1	non-cvx, fully-comp	yes	$\tilde{\mathcal{O}}\left(\varepsilon^{-2}\right)$ (5)	$\tilde{\mathcal{O}}\left(\varepsilon^{-2}\right)$ (5)	modif. LMO

Table 1: Summary of convergence complexities for solving non-smooth problems. Note (1) marks complexities reaching an ε functional residual. Note (2) marks complexities for reaching Clarke-stationary points. Note (3) marks complexities for obtaining $d-$stationary points. Note (4) marks the complexity for reaching a small norm of the gradient mapping. Finally, note (5) marks the complexity of minimizing the positive quantity (14).

Finally, two concurrent works study FW methods for some restricted classes of non-smooth and non-convex problems. De Oliveira (2023) shows that vanilla FW with line-search can be applied to the special class of upper$-C^{1,\alpha}$ functions, when one replaces gradients with an arbitrary element in the Clarke subdifferential. A rate of $\mathcal{O}\left(\varepsilon^{-2}\right)$ is shown for reaching a Clarke-stationary point in a setting comparable to ours. A similar rate is shown by Kreimeier et al. (2023) for reaching a d-stationary point of abs-smooth functions through the use of a modified LMO. Both these algorithms are structure-agnostic. A summary of method complexities for solving non-smooth problems is provided in Table 1.

Notation. We denote by $[n]$ the set $\{1, \dots n\}$ and by $\| \cdot \|$ the standard Euclidean norm, unless explicitly stated otherwise. We define the diameter of a bounded set $\mathcal{X}$ as $\mathcal{D}_{\mathcal{X}} \overset{\text{def}}{=} \max_{\mathbf{z},\mathbf{y}\in\mathcal{X}}\{\| \mathbf{z} - \mathbf{y} \|\}$. We use the notation $\Delta_n \overset{\text{def}}{=} \{\boldsymbol{\lambda} \in \mathbb{R}^n_+ \ : \ \langle \boldsymbol{\lambda}, \mathbf{e}\rangle = 1\}$ to denote the standard n-dimensional simplex, where $\mathbf{e}$ is the vector of all ones. For a differentiable, scalar-valued function $f : \mathbb{R}^d \to \mathbb{R}$ we use $\nabla f(\mathbf{x}) \in \mathbb{R}^d$ to denote its gradient vector and $\nabla^2 f(\mathbf{x}) \in \mathbb{R}^{d\times d}$ to denote its Hessian matrix. For a differentiable vector-valued function $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^n$ defined as $\mathbf{f} = (f_1, f_2, \dots f_n)$ we denote by $\nabla \mathbf{f}(\mathbf{x})$ its Jacobian matrix defined as $\nabla \mathbf{f}(\mathbf{x}) = \sum_{i=1}^n \mathbf{e}_i \nabla f_i(\mathbf{x})^\top \in \mathbb{R}^{n\times d}$, where $\mathbf{e}_i$ are the standard basis vectors in $\mathbb{R}^n$. We represent the second directional derivatives applied to the same direction $\mathbf{h} \in \mathbb{R}^d$ as $\nabla^2 f(\mathbf{x})[\mathbf{h}]^2 \overset{\text{def}}{=} \langle \nabla^2 f(\mathbf{x})\mathbf{h}, \mathbf{h}\rangle \in \mathbb{R}$, and $\nabla^2 \mathbf{f}(\mathbf{x})[\mathbf{h}]^2 \overset{\text{def}}{=} \sum_{i=1}^n \mathbf{e}_i \nabla^2 f_i(\mathbf{x})[\mathbf{h}]^2 \in \mathbb{R}^n$.

2. Problem Setup, Assumptions and Examples

The problems addressed by this work are represented by the following structured objective

$$\varphi^\star \quad = \quad \min_{\mathbf{x}\in\mathcal{X}} \Big[\varphi(\mathbf{x}) \quad \overset{\text{def}}{=} \quad F(\mathbf{f}(\mathbf{x}), \mathbf{x}) \Big], \qquad \mathcal{X} \subset \mathbb{R}^d, \tag{4}$$

where $\mathcal{X}$ is a convex and compact set and the inner mapping $\mathbf{f} : \mathcal{X} \to \mathbb{R}^n$ is differentiable and defined as $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x})) \in \mathbb{R}^n$, where each $f_i : \mathcal{X} \to \mathbb{R}$ is differentiable. We assume access to a first-order oracle $\nabla \mathbf{f}$, which is the main source of computational burden. The outer component $F : \mathbb{R}^n \times \mathcal{X} \to \mathbb{R}$, on the other hand, is *directly accessible* to the algorithm designer and is *simple* (see assumptions). However, F is possibly non-differentiable.

In this work, we propose two algorithmic solutions addressing problem (4), which we call a *fully composite problem*. Our methods importantly assume that subproblems of the form

$$\operatorname*{Argmin}_{\mathbf{x}\in\mathcal{X}} F\big(\mathbf{A}\mathbf{x}+\mathbf{b},\mathbf{x}\big) + \langle \mathbf{u},\mathbf{x}\rangle \tag{5}$$

are efficiently solvable, where $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $\mathbf{b} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^d$. Oracles of type (5) are sequentially called during the optimization procedure and take as arguments linearizations of the difficult nonlinear components of (4). Naturally, solving (5) cheaply is possible only when F is simple and $\mathcal{X}$ has an amenable structure.

In particular, template (4) encompasses to some standard problem formulations. For example, the classical Frank-Wolfe setting is recovered when $F(\mathbf{u},\mathbf{x}) \equiv u^{(1)}$, in which case problem (4) becomes $\min_{\mathbf{x}\in\mathcal{X}} f_1(\mathbf{x})$ and subproblem (5) reduces to a simple LMO: $\operatorname{argmin}_{\mathbf{x}\in\mathcal{X}}\langle\mathbf{u},\mathbf{x}\rangle$. The setting of proximal-gradient methods is similarly covered, by letting $F(\mathbf{u},\mathbf{x}) \equiv u^{(1)} + \psi(\mathbf{x})$ for a given convex function ψ (e.g., a regularizer). Then, problem (4) reduces to additive composite optimization $\min_{\mathbf{x}\in\mathcal{X}}\Big\{f_1(\mathbf{x})+\psi(\mathbf{x})\Big\}$, and subproblem (5) becomes $\operatorname{argmin}_{\mathbf{x}\in\mathcal{X}}\Big\{\langle\mathbf{u},\mathbf{x}\rangle+\psi(\mathbf{x})\Big\}$.

We now formally state the assumptions on the fully composite problem (4), along with commentary and examples.

Assumption 1 *The outer function $F : \mathbb{R}^n\times\mathcal{X} \to \mathbb{R}$ is jointly convex in its arguments. Additionally, $F(\mathbf{u},\mathbf{x})$ is subhomogeneous in $\mathbf{u}$:*

$$F(\gamma\mathbf{u},\mathbf{x}) \;\; \leq \;\; \gamma F(\mathbf{u},\mathbf{x}), \qquad \forall \mathbf{u}\in\mathbb{R}^n,\; \mathbf{x}\in\mathcal{X},\; \gamma\geq 1. \tag{6}$$

Assumption 2a *The inner mapping $\mathbf{f} : \mathcal{X} \to \mathbb{R}^n$ is differentiable and the following affine-invariant quantity is bounded:*

$$\mathcal{S} \;=\; \mathcal{S}_{\mathbf{f},F,\mathcal{X}} \;\overset{\text{def}}{=}\; \sup_{\substack{\mathbf{x},\mathbf{y}\in\mathcal{X},\,\gamma\in(0,1]\\ \mathbf{y}_\gamma=\mathbf{x}+\gamma(\mathbf{y}-\mathbf{x})}} F\Big(\tfrac{2}{\gamma^2}\big[\mathbf{f}(\mathbf{y}_\gamma)-\mathbf{f}(\mathbf{x})-\nabla\mathbf{f}(\mathbf{x})(\mathbf{y}_\gamma-\mathbf{x})\big],\mathbf{y}_\gamma\Big) \;<\; +\infty. \tag{7}$$

Assumption 2b *Each component $f_i(\cdot)$ has a Lipschitz continuous gradient on $\mathcal{X}$ with constant L_i:*

$$\|\,\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\,\| \leq L_i\,\|\,\mathbf{x}-\mathbf{y}\,\| \quad \forall \mathbf{x},\mathbf{y}\in\mathcal{X}, \forall i\in[n].$$

We denote the vector of Lipschitz constants by $\mathbf{L} = (L_1,\ldots,L_n)\in\mathbb{R}^n$.

Assumption 3 *Each component $f_i : \mathcal{X}\to\mathbb{R}$ is convex. Moreover, $F(\cdot,\mathbf{x})$ is monotone $\forall\mathbf{x}\in\mathcal{X}$. Thus, for any two vectors $\mathbf{u},\mathbf{v}\in\mathbb{R}^n$ such that $\mathbf{u}\leq\mathbf{v}$ (component-wise), it holds that*

$$F(\mathbf{u},\mathbf{x}) \;\; \leq \;\; F(\mathbf{v},\mathbf{x}). \tag{8}$$

A few comments are in order. Assumption 1, which is also required by Doikov and Nesterov (2022), represents the formal manner in which we ask that F be simple – through convexity and bounded growth in $\mathbf{u}$. This assumption ensures convexity of subproblem (5), irrespective of the nature of $\mathbf{f}$.

Assumption 2a is a generalization of the standard bounded curvature premise typical for Frank-Wolfe settings (Jaggi, 2013). Requirement (7) is mild, as it only asks that the curvature of $\mathbf{f}$ remains bounded under F over $\mathcal{X}$. Importantly, the quantity $\mathcal{S}$ is *affine-invariant* (remains unchanged under affine reparametrizations of $\mathcal{X}$), which enables us to obtain convergence rates with the same property. Further discussion on the importance of affine-invariant analysis for FW algorithms is provided by Jaggi (2013). For mappings $\mathbf{f}$ that are twice differentiable, we can bound the quantity $\mathcal{S}$ from Assumption 2a using Taylor's formula and the second derivatives, as follows

$$\mathcal{S} \quad \leq \quad \sup_{\substack{\mathbf{x},\mathbf{y}\in\mathcal{X},\,\gamma\in[0,1]\\ \mathbf{y}_\gamma=\mathbf{x}+\gamma(\mathbf{y}-\mathbf{x})}} F(\nabla^2\mathbf{f}(\mathbf{y}_\gamma)[\mathbf{y}-\mathbf{x}]^2,\mathbf{x}).$$

This quantity is reminiscent of the quadratic upper-bound used to analyze smooth optimization methods. In particular, for monotone non-decreasing F, a compact $\mathcal{X}$ and Lipschitz continuous ∇f_i with respect to a fixed norm $\|\cdot\|$, the assumption is satisfied with

$$\mathcal{S} \quad \leq \quad F(\mathbf{L}\mathcal{D}^2_{\mathcal{X}}) \quad \overset{\text{def}}{=} \quad \sup_{\mathbf{x}\in\mathcal{X}} F(\mathbf{L}\mathcal{D}^2_{\mathcal{X}},\mathbf{x}).$$

Assumption 2b is standard and considered separately from Assumption 2a to allow for different levels of generality in our results. The restriction to $\mathcal{X}$ makes this a locally-Lipschitz gradient assumption on f_i.

Finally, Assumption 3, which is also made by Doikov and Nesterov (2022), is required whenever we need to ensure the overall convexity of $\varphi(\mathbf{x})$. The monotonicity of F is necessary in addition to convexity of each f_i, since the composition of convex functions is not necessarily convex (Boyd and Vandenberghe, 2004). We rely on this assumption for deriving asymptotically faster convergence rates in the convex setting (Section 4).

To conclude this section, we provide the main application examples that fall under our fully composite template and which satisfy to our assumptions. Further examples can be found in Appendix C.

Example 1 *Let* $F(\mathbf{u},\mathbf{x}) \equiv \max\limits_{1\leq i\leq n} u^{(i)}$. *Function* F *satisfies Assumptions 1 and 3 and problem* (4) *becomes*

$$\min_{\mathbf{x}\in\mathcal{X}} \max_{1\leq i\leq n} f_i(\mathbf{x}), \tag{9}$$

while oracle (5) *becomes*

$$\min_{\mathbf{x}\in\mathcal{X}} \max_{1\leq i\leq n} \langle \mathbf{a}_i,\mathbf{x}\rangle + b_i \qquad \Leftrightarrow \qquad \min_{\mathbf{x}\in\mathcal{X},t\in\mathbb{R}} \big\{ t \;:\; \langle \mathbf{a}_i,\mathbf{x}\rangle + b_i \leq t,\, 1\leq i\leq n \big\}. \tag{10}$$

Max-type minimization problems of this kind result from scalarization approaches in multi-objective optimization, and their solutions were shown to be (weakly) Pareto optimal (Chapter 3.1 in Miettinen, 1999). As such, problem (9) *is relevant to a wide variety of applications requiring optimal trade-offs amongst several objective functions, and appears in areas such as machine learning, science and engineering (see the introductory sections of, e.g., Daulton et al., 2022; Zhang and Golovin, 2020). Problem* (9) *also covers some instances of constrained* ℓ_∞ *regression.*

When $\mathcal{X}$ *is a polyhedron, subproblem* (10) *can be solved via Linear Programming, while for general* $\mathcal{X}$ *one can resort to Interior-Point Methods (Nesterov and Nemirovski, 1994). Another option for solving* (10) *is to note that under strong duality (Rockafellar, 1970) we have*

$$\min_{\mathbf{x}\in\mathcal{X}} \max_{1\leq i\leq n} \langle \mathbf{a}_i,\mathbf{x}\rangle + b_i \quad = \quad \min_{\mathbf{x}\in\mathcal{X}} \max_{\boldsymbol{\lambda}\in\Delta_n} \sum_{i=1}^n \lambda^{(i)}\big[\langle \mathbf{a}_i,\mathbf{x}\rangle + b_i\big] = \max_{\boldsymbol{\lambda}\in\Delta_n} g(\boldsymbol{\lambda}), \tag{11}$$

where $g(\boldsymbol{\lambda}) \overset{\text{def}}{=} \min\limits_{\mathbf{x}\in\mathcal{X}} \sum_{i=1}^{n} \lambda^{(i)} \big[\langle \mathbf{a}_i, \mathbf{x}\rangle + b_i\big]$. *The maximization of* g *in* (11) *can be done very efficiently for small values of* n *(with, e.g., the Ellipsoid Method or the Mirror Descent algorithm), since evaluating* $g(\boldsymbol{\lambda})$ *and* $\partial g(\boldsymbol{\lambda})$ *reduces to a vanilla LMO call over* $\mathcal{X}$. *An interesting case is* $n = 2$, *for which* (11) *becomes a* univariate *maximization problem and one may use binary search to solve it at the expense of a logarithmic number of LMOs.*

Example 2 *Let* $F(\mathbf{u}, \mathbf{x}) \equiv \|\mathbf{u}\|$ *for an arbitrary fixed norm* $\|\cdot\|$. *Function* F *satisfies Assumption 1 and problem* (4) *can be interpreted as solving a system of non-linear equations over* $\mathcal{X}$

$$\min_{\mathbf{x}\in\mathcal{X}} \|\mathbf{f}(\mathbf{x})\|, \tag{12}$$

while oracle (5) *amounts to solving the (constrained) linear system* $\min\limits_{\mathbf{x}\in\mathcal{X}} \|\mathbf{A}\mathbf{x} + \mathbf{b}\|$. *Problems of this kind can be encountered in applications such as robust phase retrieval (Duchi and Ruan, 2019) with phase constraints.*

The iterations of Algorithm 1 can be interpreted as a variant of the Gauss-Newton method (Burke and Ferris, 1995; Nesterov, 2007; Tran-Dinh et al., 2020), solving the (constrained) linear systems:

$$\mathbf{x}_{k+1} \in \operatorname*{Argmin}_{\mathbf{x}\in\mathcal{X}} \|\mathbf{f}(\mathbf{y}_k) + \nabla \mathbf{f}(\mathbf{y}_k)(\mathbf{x} - \mathbf{y}_k)\|, \quad \textit{and} \quad \mathbf{y}_{k+1} = (1-\gamma_k)\mathbf{y}_k + \gamma_k \mathbf{x}_{k+1}. \tag{13}$$

In the particular case of solving systems of non-linear equations over compact convex sets, our algorithms can be seen as modified Gauss-Newton methods with global convergence guarantees.

3. The Basic Method

We present the first new method for solving problem (4) in Algorithm 1. The central idea is to *linearize* the differentiable components of the objective and then to minimize this new model over the constraint $\mathcal{X}$, via calls to an oracle of type (5). The next iterate is defined as a convex combination with coefficient (or *stepsize*) γ between the computed minimizer and the preceding iterate.

Algorithm 1 Basic Method

```
Input: $\mathbf{y}_0 \in \mathcal{X}$
for $k = 0, 1, \ldots$ do
    Compute
        $\mathbf{x}_{k+1} \in \operatorname*{Argmin}_{\mathbf{x}\in\mathcal{X}} F\big(\mathbf{f}(\mathbf{y}_k) + \nabla \mathbf{f}(\mathbf{y}_k)(\mathbf{x} - \mathbf{y}_k),\, \mathbf{x}\big)$
    Choose $\gamma_k \in (0, 1]$ by a predefined rule or with line search
    Set $\mathbf{y}_{k+1} = (1-\gamma_k)\mathbf{y}_k + \gamma_k \mathbf{x}_{k+1}$
end for
```

A similar method for tackling problems of type (4) in the convex setting was proposed by Doikov and Nesterov (2022). Different from theirs, our method decouples the parameter γ_k from the minimization subproblem. This change is crucial since it allows us to choose the parameter γ_k *after* minimizing the model, thus enabling us to use efficient line search rules. Moreover, we provide

Algorithm 1 with a more advanced *affine-invariant* analysis and establish its convergence in the *non-convex* setup.

We also mention that for solving problems of type (9), oracle (5) reduces to the minimization of a piecewise linear function over $\mathcal{X}$. Therefore, it has the same complexity as the modified LMOs of Kreimeier et al. (2023) and, moreover, subproblem (5) is convex irrespective of the nature of $\mathbf{f}$.

Accuracy Certificates. The standard *progress metric* of FW algorithms, which Algorithm 1 generalizes, is the 'Frank-Wolfe gap' or Hearn gap (Hearn, 1982). For smooth objectives, it is defined as $\Delta_k = \max_{\mathbf{y} \in \mathcal{X}} \langle \nabla \varphi(\mathbf{y}_k), \mathbf{y}_k - \mathbf{y} \rangle$, for each iterate $\mathbf{y}_k$. This quantity is computed cost-free during the algorithm's iterations and has the desirable property of upper-bounding the suboptimality of the current iterate: $\Delta_k \geq \varphi(\mathbf{y}_k) - \varphi^\star$. Notably, its semantics straightforwardly extend to the non-convex setting (Lacoste-Julien, 2016). Additionally, convergence guarantees on the gap are desirable due to its affine invariance, which aligns with the affine invariance of classical FW algorithm.

Our setting does not permit a direct generalization of the FW gap with all of the above properties. Rather, we introduce the following *accuracy certificate*, which is readily available in each iteration:

$$\Delta_k \overset{\text{def}}{=} \varphi(\mathbf{y}_k) - F\big(\mathbf{f}(\mathbf{y}_k) + \nabla \mathbf{f}(\mathbf{y}_k)(\mathbf{x}_{k+1} - \mathbf{y}_k), \mathbf{x}_{k+1}\big). \tag{14}$$

For minimization of a smooth (not necessarily convex) function, quantity (14) indeed reduces to the standard FW gap. Moreover, for convex $\varphi(\mathbf{x})$ (Assumption 3) we can conclude that

$$\begin{aligned} \Delta_k &\geq \max_{\mathbf{x} \in \mathcal{X}} \Big[\varphi(\mathbf{y}_k) - F\big(\mathbf{f}(\mathbf{y}_k) + \nabla \mathbf{f}(\mathbf{y}_k)(\mathbf{x} - \mathbf{y}_k), \mathbf{x}\big) \Big] \\ &\geq \max_{\mathbf{x} \in \mathcal{X}} \Big[\varphi(\mathbf{y}_k) - F(\mathbf{f}(\mathbf{x}), \mathbf{x}) \Big] \;=\; \varphi(\mathbf{y}_k) - \varphi^\star. \end{aligned} \tag{15}$$

Hence, for a tolerance $\varepsilon > 0$, the criterion $\Delta_k \leq \varepsilon$ can be used as the stopping condition for our method in convex scenarios. Moreover, the value of Δ_k can be used for computing the parameter γ_k through line search.

Convergence on Convex Problems. In the following, we prove the global convergence of Algorithm 1 in case when $\varphi(\mathbf{x})$ is convex.

Theorem 3.1 *Let Assumptions 1, 2a, and 3 be satisfied. Let* $\gamma_k := \min\{1, \frac{\Delta_k}{\mathcal{S}}\}$ ***or*** $\gamma_k := \frac{2}{2+k}$. *Then, for* $k \geq 1$ *it holds that*

$$\varphi(\mathbf{y}_k) - \varphi^\star \;\leq\; \frac{2\mathcal{S}}{1+k} \qquad \textit{and} \qquad \min_{1 \leq i \leq k} \Delta_i \;\leq\; \frac{6\mathcal{S}}{k}. \tag{16}$$

Our method recovers the rate of classical FW methods for smooth problems, while being applicable to the wider class of *fully composite problems* (4). Thus, our $\mathcal{O}(1/k)$ rate improves upon the $\mathcal{O}(1/\sqrt{k})$ of black-box non-smooth optimization. Clearly, the improvement is achievable by leveraging the *structure of the objective* within the algorithm.

Convergence on Non-convex Problems. In this case, Δ_k has different semantics and no longer provides an accuracy certificate for the functional residual. This quantity is nevertheless important, since it enables us to quantify the algorithm's progress in the non-convex setting, while maintaining an affine-invariant analysis. The following theorem states the convergence guarantee on Δ_k for non-convex problems.

Theorem 3.2 *Let Assumptions 1 and 2a be satisfied. Let $\gamma_k := \min\{1, \frac{\Delta_k}{\mathcal{S}}\}$ **or** $\gamma_k := \frac{1}{\sqrt{1+k}}$. Then, for all $k \geq 1$ it holds that*

$$\min_{0 \leq i \leq k} \Delta_i \quad \leq \quad \tfrac{\varphi(\mathbf{y}_0) - \varphi^\star + 0.5\mathcal{S}(1+\ln(k+1))}{\sqrt{k+1}}. \tag{17}$$

Theorem 3.2 recovers a similar rate to the classical FW methods (Lacoste-Julien, 2016). The line search rule for parameter γ_k makes our method universal, thereby allowing us to attain practically faster rates automatically when the iterates lie within a *convex region* of the objective.

As previously mentioned, the progress measure (14) does not upper-bound functional suboptimality in the general non-convex setting. However, in some cases, we may still be able to establish convergence of meaningful quantities for non-convex fully composite problems with the linearization method. Namely, let us consider problem (12) in Example 2 for the Euclidean norm, i.e., $F(\mathbf{u}, \mathbf{x}) = \| \, \mathbf{u} \, \|$, and the following simple iterations:

$$\mathbf{y}_{k+1} \quad \in \quad \operatorname*{Argmin}_{\mathbf{y} \in \mathbf{y}_k + \gamma_k(\mathcal{X} - \mathbf{y}_k)} \|\mathbf{f}(\mathbf{y}_k) + \nabla \mathbf{f}(\mathbf{y}_k)(\mathbf{y} - \mathbf{y}_k)\|. \tag{18}$$

Note that in (18), differently from (13), the value of γ_k is selected prior to the oracle call. Denoting the squared objective as $\Phi(\mathbf{x}) \overset{\text{def}}{=} \frac{1}{2}\big[\varphi(\mathbf{x})\big]^2 = \frac{1}{2}\|\mathbf{f}(\mathbf{x})\|^2$ and following our analysis, we can state the convergence of process (18) in terms of the classical FW gap with respect to Φ. The proof is deferred to Appendix A.5.

Proposition 3.1 *Let $\gamma_k := \frac{1}{\sqrt{1+k}}$. Then, for the iterations (18), under Assumption 2b and for all $k \geq 1$, it holds that*

$$\min_{0 \leq i \leq k} \max_{\mathbf{y} \in \mathcal{X}} \langle \nabla \Phi(\mathbf{y}_i), \mathbf{y}_i - \mathbf{y} \rangle \quad \leq \quad \mathcal{O}\big(\tfrac{\ln(k)}{\sqrt{k}}\big).$$

We further show in Appendix B that Δ_k can be related to the classical FW gap, when our iterates lie in a smooth region of F. Whether we can provide a meaningful interpretation of Δ_k in the general non-convex case, however, remains an interesting open question.

4. The Accelerated Method

We now move away from the affine-invariant formulation of Algorithm 1 to a setting in which, by considering regularized minimization subproblems along with convexity and Lipschitz continuity of gradients, we can *accelerate* the Basic Method. We achieve acceleration by resorting to the well-known three-point scheme of Nesterov (1983), in which the proximal subproblem is solved inexactly via calls to oracles of type (5). This approach was first analyzed in the context of FW methods by Lan and Zhou (2016).

We propose Algorithm 2 which consists of a two-level scheme: an outer-loop computing the values of three iterates $\mathbf{y}$, $\mathbf{x}$ and $\mathbf{z}$ in $\mathcal{X}$, and a subsolver computing inexact solutions to the *proximal subproblem*

$$\operatorname*{Argmin}_{\mathbf{u} \in \mathcal{X}} \Big\{ P(\mathbf{u}) \overset{\text{def}}{=} F(\mathbf{f}(\mathbf{z}) + \nabla \mathbf{f}(\mathbf{z})(\mathbf{u} - \mathbf{z}), \mathbf{u}) + \tfrac{\beta}{2}\|\mathbf{u} - \mathbf{x}\|_2^2, \;\; \beta > 0 \Big\}. \tag{19}$$

Note that the minimization in (19) does not conform to our oracle model (5) due to the quadratic regularizer. However, we can approximate its solution by iteratively solving subproblems in which we linearize the squared norm to match the template of (5). This procedure, denoted as InexactProx in Algorithm 2, returns a point $\mathbf{u}^+$ satisfying the optimality condition *η-inexactly* for some $\eta > 0$:

$$\begin{aligned} &F(\mathbf{f}(\mathbf{z}) + \nabla f(\mathbf{z})(\mathbf{u}^+ - \mathbf{z}), \mathbf{u}^+) + \beta\langle \mathbf{u}^+ - \mathbf{x}, \mathbf{u}^+\rangle \\ &\qquad \le F(\mathbf{f}(\mathbf{z}) + \nabla f(\mathbf{z})(\mathbf{u} - \mathbf{z}), \mathbf{u}) + \beta\langle \mathbf{u}^+ - \mathbf{x}, \mathbf{u}\rangle + \eta, \qquad \forall \mathbf{u} \in \mathcal{X}. \end{aligned} \tag{20}$$

Note that condition (20) implies $P(\mathbf{u}^+) \le P(\mathbf{u}) + \eta,\ \forall \mathbf{u} \in \mathcal{X}$. Formally, the main convergence result characterizing Algorithm 2 is the following.

Algorithm 2 Accelerated Method

```
Input: $\mathbf{y}_0 \in \mathcal{X}$, set $\mathbf{x}_0 = \mathbf{y}_0$
for $k = 0, 1, \dots$ do
    Choose $\gamma_k \in (0, 1]$
    Set $\mathbf{z}_{k+1} = (1 - \gamma_k)\mathbf{y}_k + \gamma_k \mathbf{x}_k$
    Compute $\mathbf{x}_{k+1} = \text{InexactProx}(\mathbf{x}_k, \mathbf{z}_{k+1}, \beta_k, \eta_k)$ for some $\beta_k \ge 0$ and $\eta_k \ge 0$
    Set $\mathbf{y}_{k+1} = (1 - \gamma_k)\mathbf{y}_k + \gamma_k \mathbf{x}_{k+1}$
end for
```

Theorem 4.1 *Let Assumptions 1, 2b, and 3 be satisfied. We choose $\gamma_k := \frac{3}{k+3}$, $\beta_k := cF(\mathbf{L})\gamma_k$ and $\eta_k := \frac{\delta}{3(k+1)(k+2)}$ where $\delta > 0$ and $c \ge 0$ are chosen constants, and $F(\mathbf{L}) := \sup_{\mathbf{x} \in \mathcal{X}} F(\mathbf{L}, \mathbf{x})$. Then, for all $k \ge 1$ it holds that*

$$\varphi(\mathbf{y}_k) - \varphi^\star \quad \le \quad \frac{\delta + 8cF(\mathbf{L})\mathcal{D}_{\mathcal{X}}^2}{(k+2)(k+3)} + \frac{2\max\{0, 1-c\}F(\mathbf{L})\mathcal{D}_{\mathcal{X}}^2}{k+3}.$$

The proof of Theorem 4.1 comes from a natural sequence of steps involving the properties of the operators and the approximate optimality of $\mathbf{x}_{k+1}$. The crucial step in attaining the improved convergence is the choice of parameters γ_k, β_k and η_k. Notably, the decay speed required of η_k is quadratic, meaning that the subproblems are solved with fast-increasing accuracy and at the cost of additional time spent in the subsolver. The constant δ allows us to fine-tune the accuracy required for the first several iterations of the algorithm, where we can demand a lower accuracy. In practice, we can always choose $\delta = 1$ as a universal rule, and the optimal choice is $\delta = F(\mathbf{L})\mathcal{D}_{\mathcal{X}}^2$ when these parameters are known. The factor $cF(\mathbf{L})$ in the definition of β_k can be interpreted as the quality of the approximation of the Lipschitz constant for our problem. Namely, it is exactly computed for $c = 1$, and over or underestimated for $c > 1$ and $c \in (0, 1)$ respectively.

We describe each of the bounding terms independently: the first is highly reminiscent of the usual bounds accompanying FW-type algorithms in terms of constants, albeit now with quadratic decay speed. The second term indicates the behavior of the algorithm as a function of c: overestimation of $F(\mathbf{L})$ ensures quadratic rates of convergence, since the second term becomes negative. Conversely, underestimation of $F(\mathbf{L})$ brings us back into the familiar FW convergence regime of $\mathcal{O}(1/k)$ as the second term becomes positive. The extreme case $c = 0$ (and hence $\beta_k = 0$) essentially reduces Algorithm 2 to Algorithm 1, since the projection subproblem reduces to problem (5) which we assume to be easily solvable. We therefore have robustness in terms of choosing the parameter c and the exact knowledge of $F(\mathbf{L})$ is not needed, even though it may come at the cost of a slower convergence. In contrast, classical Fast Gradient Methods are usually very sensitive to such parameter choices (Devolder, 2013).

Theorem 4.1 provides an accelerated rate on the iterates $\mathbf{y}_k$ – an analogous result to that of Lan and Zhou (2016) albeit under a different oracle. This convergence rate is conditioned on the subsolver returning an η_k-inexact solution to the projection subproblem and therefore any subsolver satisfying the condition can achieve this rate. As with any optimization algorithm, convergence guarantees may also be stated in terms of the oracle complexity required to reach ε accuracy. For Algorithm 2 all the oracle calls are deferred to the subsolver InexactProx, which we describe and analyze in the next section.

5. Solving the Proximal Subproblem

We now provide an instance of the InexactProx subsolver which fully determines the oracle complexity of the Accelerated Method (Algorithm 2). It relies on a specific adaptation of Algorithm 1 to the structure of (19). The quadratic regularizer is linearized and oracles of type (5) are called once per inner iteration, while the Jacobian $\nabla\mathbf{f}(\mathbf{z}_k)$ is computed once per subsolver call. The main challenge here is to find a readily available quantity defining the exit condition of the subsolver, which we denote by Δ_t.

Algorithm 3 InexactProx($\mathbf{x}$, $\mathbf{z}$, β, η)

```
Initialization: $\mathbf{u}_0 = \mathbf{x}$.
for $t = 0, 1, \ldots$ do
    Compute $\mathbf{v}_{t+1} \in \operatorname*{Argmin}_{\mathbf{v}\in\mathcal{X}} \Big\{ F\big(\mathbf{f}(\mathbf{z}) + \nabla\mathbf{f}(\mathbf{z})(\mathbf{v}-\mathbf{z}),\, \mathbf{v}\big) \;+\; \beta\langle \mathbf{u}_t - \mathbf{x}, \mathbf{v}\rangle \Big\}$
    Compute $\Delta_t \;=\; F\big(\mathbf{f}(\mathbf{z}) + \nabla\mathbf{f}(\mathbf{z})(\mathbf{u}_t-\mathbf{z}),\, \mathbf{u}_t\big) \;-\; F\big(\mathbf{f}(\mathbf{z}) + \nabla\mathbf{f}(\mathbf{z})(\mathbf{v}_{t+1}-\mathbf{z}),\, \mathbf{v}_{t+1}\big)$
                    $+\; \beta\langle \mathbf{u}_t - \mathbf{x}, \mathbf{u}_t - \mathbf{v}_{t+1}\rangle$
    if $\Delta_t \le \eta$ then return $\mathbf{u}_t$
    Set $\alpha_t = \min\Big\{1, \frac{\Delta_t}{\beta\|\mathbf{v}_{t+1}-\mathbf{u}_t\|_2^2}\Big\}$ and $\mathbf{u}_{t+1} = \alpha_t\mathbf{v}_{t+1} + (1-\alpha_t)\mathbf{u}_t$
end for
```

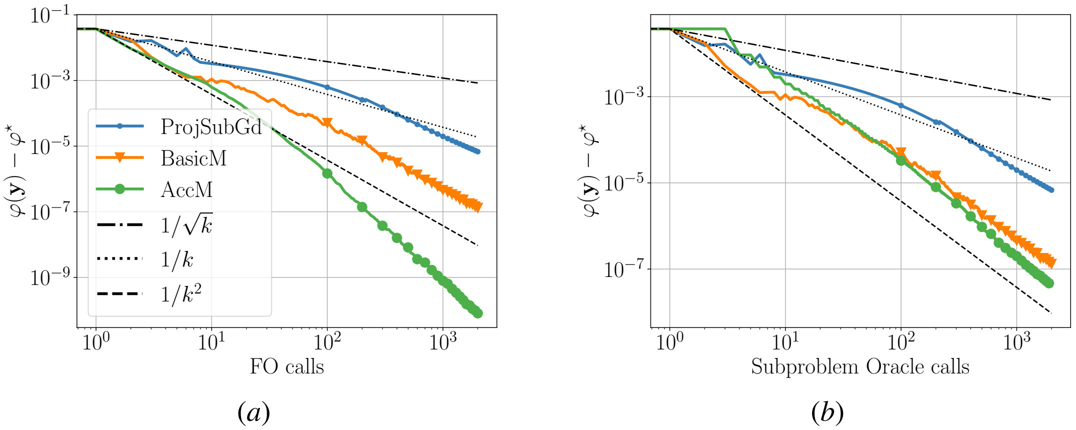
The parameters of Algorithm 3 are fully specified, and the stopping condition depends on $\Delta_t \geq P(\mathbf{u}_t) - P^\star$, which is a meaningful progress measure. The algorithm selects its stepsize via closed-form line search to improve practical performance. When $F(\mathbf{u}) \equiv \mathbf{u}^{(1)}$, this procedure recovers the classical FW algorithm with line search applied to problem (19).

We prove two results in relation to Algorithm 3: its convergence rate and the total oracle complexity of Algorithm 2 when using Algorithm 3 as the subsolver. The rate and analysis are similar to the ones for the Basic Method, utilizing additionally the form of the proximal subproblem.

Theorem 5.1 *Let Assumptions 1, 2b, and 3 be satisfied. Then, for all $t \geq 1$ it holds that*

$$P(\mathbf{u}_t) - P^\star \;\leq\; \tfrac{2\beta\mathcal{D}_{\mathcal{X}}^2}{t+1} \qquad \textit{and} \qquad \min_{1\leq i\leq t} \Delta_t \;\leq\; \tfrac{6\beta\mathcal{D}_{\mathcal{X}}^2}{t}.$$

Consequently, Algorithm 3 returns an η-approximate solution according to condition (20) after at most $\mathcal{O}\Big(\frac{\beta\mathcal{D}_{\mathcal{X}}^2}{\eta}\Big)$ iterations.



(*a*) (*b*)

Figure 1: Convergence of the Basic and Accelerated methods against the Projected Subgradient baseline for problem (21), along with relevant theoretical rates.

We note that oracle (5) is called once per inner iteration and the Jacobian $\nabla \mathbf{f}(\mathbf{z}_k)$ is computed once per subsolver call. In particular, when using Algorithm 3 as a subsolver, our Accelerated Method achieves the optimal number of $\mathcal{O}\left(\varepsilon^{-1/2}\right)$ Jacobian computations typical of smooth and convex optimization, while maintaining a $\mathcal{O}\left(\varepsilon^{-1}\right)$ complexity for the number of calls to oracle (5). The results are stated in the following corollary.

Corollary 5.1 *Consider the optimal choice of parameters for Algorithm 2, that is $c := 1$ and $\delta := F(\mathbf{L})\mathcal{D}^2_{\mathcal{X}}$. Then, solving problem (4) with ε accuracy $\varphi(\mathbf{y}_k) - \varphi^\star \leq \varepsilon$, requires $\mathcal{O}\Big(\sqrt{\frac{F(\mathbf{L})\mathcal{D}^2_{\mathcal{X}}}{\varepsilon}}\Big)$ computations of $\nabla \mathbf{f}$. In addition, the total number of calls to oracle (5) is $\mathcal{O}\big(\frac{F(\mathbf{L})\mathcal{D}^2_{\mathcal{X}}}{\varepsilon}\big)$.*

Finally, we note that for smaller values of parameter $c \in [0, 1]$ in Algorithm 2 (underestimating the Lipschitz constant), the complexity of InexactProx procedure improves. Thus, for $c = 0$ we have $\beta = 0$ (no regularization) and Algorithm 3 finishes after just *one step*.

6. Experiments

The experiments are implemented in `Python 3.9` and run on a MacBook Pro M1 with 16 GB RAM. For both experiments we use the Projected Subgradient Method as a baseline (Shor et al., 1985), with a stepsize of $\frac{p}{\sqrt{k}}$ where p is tuned for each experiment. The `CVXPY` library (Diamond and Boyd, 2016) is used to solve subproblems of type (5). The random seed for our experiments is always set to `666013`, and we set $c = 1$ since we can analytically compute the Lipschitz constants or their upper bounds.

6.1. Minimization Over the Simplex

We consider the following optimization problem

$$\min_{\mathbf{x}\in\mathcal{X}} \left\{ \max_{i=1,n} \mathbf{x}^\top \mathbf{A}_i \mathbf{x} - \mathbf{b}_i^\top \mathbf{x} \right\}, \text{ for } \mathcal{X} = \Delta_d, \subseteq \mathbb{R}^d, \tag{21}$$

where $\mathbf{A}_i \in \mathbb{R}^{d\times d}$ are random PSD matrices and $\mathbf{b} \in \mathbb{R}^d$. The problem conforms to Example 1, and we use $d = 500$ and $n = 10$. We generate $\mathbf{A}_i = \mathbf{Q}_i \mathbf{D} \mathbf{Q}_i^\top$, where $\mathbf{D}$ is a diagonal matrix of

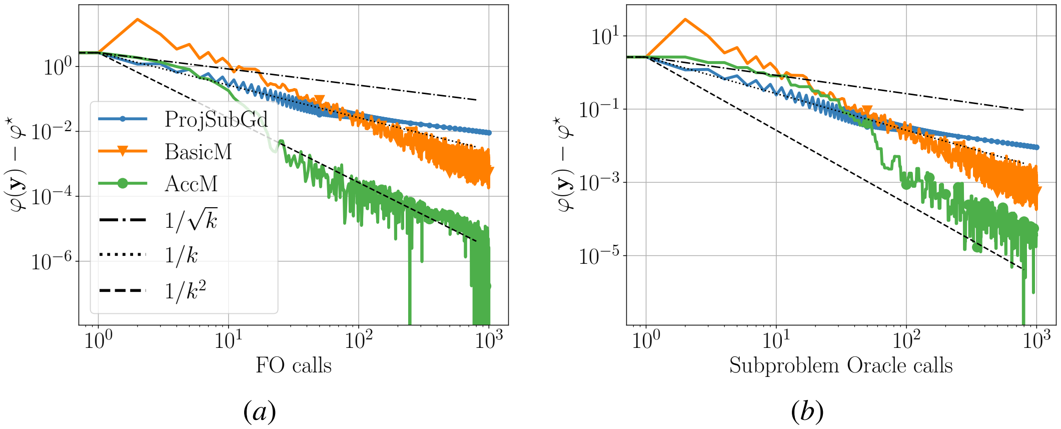


Figure 2: Convergence of the Basic and Accelerated methods against the Projected Subgradient baseline for problem (22), along with relevant theoretical rates.

eigenvalues decaying linearly from 1 to 10^{-6}, and $\mathbf{Q}_i$ is a randomly generated orthogonal matrix using the `scipy.stats.ortho_group` method (Mezzadri, 2006). The vectors $\mathbf{b}_i$, which determine the position of the quadratics in space, are set as follows: $\mathbf{b}_i = \mathbf{e}_i \cdot 10$, $\mathbf{b}_9 = \mathbf{0}$ (the origin), $\mathbf{b}_{10} = \mathbf{1} \cdot 10$ (the all ones vector multiplied by 10). We set $\delta = 0.2$ in the Accelerated Method (see Theorem 4.1) and settle for $p = 1.42$ following tuning of the Subgradient Method. Finally, we set $\mathbf{x}_0 = \mathbf{e}_3 \in \Delta_d$ for all methods.

The convergence results in terms of FO oracles and oracles of type (5) are shown in Figure 1(*a*) and 1(*b*), respectively. The figures highlight the improvement in terms of the number of FO calls, while showing comparable performance in terms of subproblem oracle calls, as predicted by our theory.

6.2. Minimization Over the Nuclear Norm Ball.

We consider the following optimization problem

$$\min_{\mathbf{X}\in\mathcal{X}} \left\{ \max_{i=1,n} \sum_{(k,l)\in\Omega_i} \left(\mathbf{X}_{k,l} - \mathbf{A}^{(i)}_{k,l} \right)^2 \right\}, \text{ for } \mathcal{X} := \{\mathbf{X} \in \mathbb{R}^{d\times m},\ \|\, \mathbf{X} \,\|_* \le r\} \tag{22}$$

Formulation (22) models a matrix completion scenario where we wish to recover an $\mathbf{X}^\star$ that minimizes the largest error within a given *set* of matrices $\mathbf{A}^{(i)}$. The matrices $\mathbf{A}^{(i)}$ are only partially revealed through a set of corresponding indices Ω_i. This problem conforms to Example 1 and we use $d = 30$, $m = 10$, $r = 7$, where r is the rank of matrices $\mathbf{A}^{(i)}$. The data is generated in identical fashion as in Section 5.2 of (Lan and Zhou, 2016) on Matrix Completion. We set $\delta = 100$ in the Accelerated Method (see Theorem 4.1) and settle for $p = 0.2$ following tuning of the Subgradient Method. Finally, we set $\mathbf{x}_0 = \mathbf{0}^{d\times m} \in \mathcal{X}$ for all methods.

The convergence results in terms of FO oracles and oracles of type (5) are shown in Figure 2(*a*) and 2(*b*), respectively. The figures highlight the improvement in terms of the number of FO calls, while showing comparable performance in terms of subproblem oracle calls, as predicted by our theory.

7. Conclusion

Our work illustrates how improved convergence rates may be attained by assuming precise structure within a class of objectives (e.g., non-differentiable ones). Moreover, it shows how a simple principle such as linearizing the differentiable components of a function composition can be used to create more benign subproblems that are efficiently solved. Interesting future work may address relaxing the assumptions on F, extending this framework to stochastic settings, and meaningfully interpreting the quantity Δ_k in the non-convex setting.

Acknowledgments

The authors thank Aditya Varre and the anonymous reviewers for their helpful feedback on the manuscript. This work was partly supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number 22.00133.

References

Andreas Argyriou, Marco Signoretto, and Johan Suykens. Hybrid conditional gradient-smoothing algorithms with applications to sparse and low rank regularization. *Regularization, Optimization, Kernels, and Support Vector Machines*, pages 53–82, 2014.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

Jérôme Bolte, Zheng Chen, and Edouard Pauwels. The multiproximal linearization method for convex composite problems. *Mathematical Programming*, 182(1):1–36, 2020.

Radu Ioan Boţ, Sorin-Mihai Grad, and Gert Wanka. New constraint qualification and conjugate duality for composed convex optimization problems. *Journal of Optimization Theory and Applications*, 135:241–255, 2007.

Radu Ioan Boţ, Sorin-Mihai Grad, and Gert Wanka. A new constraint qualification for the formula of the subdifferential of composed convex functions in infinite dimensional spaces. *Mathematische Nachrichten*, 281(8):1088–1107, 2008.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Gábor Braun, Alejandro Carderera, Cyrille W Combettes, Hamed Hassani, Amin Karbasi, Aryan Mokhtari, and Sebastian Pokutta. Conditional gradient methods. *arXiv preprint arXiv:2211.14103*, 2022.

James V Burke. Descent methods for composite nondifferentiable optimization problems. *Mathematical Programming*, 33(3):260–279, 1985.

James V Burke. Second order necessary and sufficient conditions for convex composite ndo. *Mathematical programming*, 38:287–302, 1987.

James V Burke and Michael C Ferris. A Gauss—Newton method for convex composite optimization. *Mathematical Programming*, 71(2):179–194, 1995.

James V Burke, Hoheisel Tim, and Quang V Nguyen. A study of convex convex-composite functions via infimal convolution with applications. *Mathematics of Operations Research*, 46(4): 1324–1348, 2021.

Alejandro Carderera, Jelena Diakonikolas, Cheuk Yin Lin, and Sebastian Pokutta. Parameter-free locally accelerated conditional gradients. *arXiv preprint arXiv:2102.06806*, 2021.

Zhaoyue Chen and Yifan Sun. Accelerating frank-wolfe via averaging step directions. *arXiv preprint arXiv:2205.11794*, 2022.

Cyrille Combettes and Sebastian Pokutta. Boosting frank-wolfe by chasing gradients. In *International Conference on Machine Learning*, pages 2111–2121. PMLR, 2020.

Cyrille W Combettes and Sebastian Pokutta. Complexity of linear minimization and projection on some sets. *Operations Research Letters*, 2021.

Ying Cui, Jong-Shi Pang, and Bodhisattva Sen. Composite difference-max programs for modern statistical estimation problems. *SIAM Journal on Optimization*, 28(4):3344–3374, 2018.

Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Multi-objective bayesian optimization over high-dimensional search spaces. In *Uncertainty in Artificial Intelligence*, pages 507–517. PMLR, 2022.

Welington De Oliveira. Short paper-a note on the frank–wolfe algorithm for a class of nonconvex and nonsmooth optimization problems. *Open Journal of Mathematical Optimization*, 4:1–10, 2023.

Olivier Devolder. *Exactness, inexactness and stochasticity in first-order methods for large-scale convex optimization*. PhD thesis, ICTEAM and CORE, Université catholique de Louvain, 2013.

Jelena Diakonikolas, Alejandro Carderera, and Sebastian Pokutta. Locally accelerated conditional gradients. In *International Conference on Artificial Intelligence and Statistics*, pages 1737–1747. PMLR, 2020.

Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

Nikita Doikov and Yurii Nesterov. High-order optimization methods for fully composite problems. *SIAM Journal on Optimization*, 32(3):2402–2427, 2022.

Dmitriy Drusvyatskiy and Adrian S Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *Mathematics of Operations Research*, 43(3):919–948, 2018.

Dmitriy Drusvyatskiy and Courtney Paquette. Efficiency of minimizing compositions of convex functions and smooth maps. *Mathematical Programming*, 178(1):503–558, 2019.

John C Duchi and Feng Ruan. Solving (most) of a set of quadratic equalities: Composite optimization for robust phase retrieval. *Information and Inference: A Journal of the IMA*, 8(3):471–529, 2019.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

Dan Garber and Elad Hazan. A linearly convergent variant of the conditional gradient algorithm under strong convexity, with applications to online and stochastic optimization. *SIAM Journal on Optimization*, 26(3):1493–1528, 2016.

Donald W. Hearn. The gap function of a convex program. *Operations Research Letters*, 1(2):67–71, apr 1982. doi: 10.1016/0167-6377(82)90049-9.

Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*, pages 427–435, 2013.

Timo Kreimeier, Sebastian Pokutta, Andrea Walther, and Zev Woodstock. On a frank-wolfe approach for abs-smooth functions. *arXiv preprint arXiv:2303.09881*, 2023.

Simon Lacoste-Julien. Convergence rate of Frank-Wolfe for non-convex objectives. *arXiv preprint arXiv:1607.00345*, 2016.

Guanghui Lan. The complexity of large-scale convex programming under a linear optimization oracle. *arXiv preprint arXiv:1309.5550*, 2013.

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

Claude Lemaréchal. Cauchy and the gradient method. *Doc Math Extra*, 251(254):10, 2012.

Francesco Mezzadri. How to generate random matrices from the classical compact groups. *arXiv preprint math-ph/0609050*, 2006.

Kaisa Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media, 1999.

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

Arkadi Nemirovski and David Yudin. Problem complexity and method efficiency in optimization. 1983.

Yurii Nesterov. A method for solving the convex programming problem with convergence rate O(1/kˆ2). In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983.

Yurii Nesterov. Effective methods in nonlinear programming. *Moscow, Radio i Svyaz*, 1989.

Yurii Nesterov. Modified Gauss–Newton scheme with worst case guarantees for global performance. *Optimisation Methods and Software*, 22(3):469–483, 2007.

Yurii Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

Yurii Nesterov. Complexity bounds for primal-dual methods minimizing the model of objective function. *Mathematical Programming*, 171(1):311–330, 2018a.

Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018b.

Yurii Nesterov and Arkadi Nemirovski. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

Teemu Pennanen. Graph-convex mappings and k-convex functions. *Journal of Convex Analysis*, 6 (2):235–266, 1999.

Chao Qu, Yan Li, and Huan Xu. Non-convex conditional gradient sliding. In *international conference on machine learning*, pages 4208–4217. PMLR, 2018.

Sathya N Ravi, Maxwell D Collins, and Vikas Singh. A deterministic nonsmooth frank wolfe algorithm with coreset guarantees. *Informs Journal on Optimization*, 1(2):120–142, 2019.

R Tyrrell Rockafellar. *Convex analysis*, volume 36. Princeton university press, 1970.

NZ Shor, Krzysztof C Kiwiel, and Andrzej Ruszcayński. Minimization methods for non-differentiable functions, 1985.

Kiran K Thekumparampil, Prateek Jain, Praneeth Netrapalli, and Sewoong Oh. Projection efficient subgradient method and optimal nonsmooth frank-wolfe method. *Advances in Neural Information Processing Systems*, 33:12211–12224, 2020.

Quoc Tran-Dinh, Nhan Pham, and Lam Nguyen. Stochastic Gauss-Newton algorithms for non-convex compositional optimization. In *International Conference on Machine Learning*, pages 9572–9582. PMLR, 2020.

Alp Yurtsever, Olivier Fercoq, Francesco Locatello, and Volkan Cevher. A conditional gradient framework for composite convex minimization with applications to semidefinite programming. In *International Conference on Machine Learning*, pages 5727–5736. PMLR, 2018.

Alp Yurtsever, Suvrit Sra, and Volkan Cevher. Conditional gradient methods via stochastic path-integrated differential estimator. In *International Conference on Machine Learning*, pages 7282–7291. PMLR, 2019.

Richard Zhang and Daniel Golovin. Random hypervolume scalarizations for provable multi-objective black box optimization. In *International Conference on Machine Learning*, pages 11096–11105. PMLR, 2020.

Renbo Zhao and Robert M Freund. Analysis of the frank–wolfe method for convex composite optimization involving a logarithmically-homogeneous barrier. *Mathematical Programming*, pages 1–41, 2022.

Appendix A. Proofs

Assumption 2a implies global progress bounds on our fully composite objective with an inner linearization of $\mathbf{f}$, as stated in the following Lemma A.1. This lemma provides a basis for all our convergence results.

Lemma A.1 *Let* $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ *and* $\gamma \in [0,1]$. *Denote* $\mathbf{y}_\gamma = \mathbf{x} + \gamma(\mathbf{y} - \mathbf{x})$. *Then, it holds*

$$\varphi(\mathbf{y}_\gamma) \;\; \leq \;\; F\big(\mathbf{f}(x) + \nabla\mathbf{f}(\mathbf{x})(\mathbf{y}_\gamma - \mathbf{x}),\, \mathbf{y}_\gamma\big) \; + \; \tfrac{\gamma^2}{2}\mathcal{S}. \tag{23}$$

Proof Note that the subhomogenity assumption (6) is equivalent to the following useful inequality for the outer component of the objective see (Theorem 7.1 in Doikov and Nesterov (2022)):

$$F(\mathbf{u} + t\mathbf{v}, \mathbf{x}) \;\; \leq \;\; F(\mathbf{u}, \mathbf{x}) + tF(\mathbf{v}, \mathbf{x}), \qquad \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^n,\; \mathbf{x} \in X,\; t \geq 0. \tag{24}$$

Then, we have

$$\begin{aligned}
&\varphi(\mathbf{y}_\gamma) \;\; \equiv \;\; F(\mathbf{f}(\mathbf{y}_\gamma), \mathbf{y}_\gamma) \\
&\;= \; F\big(\mathbf{f}(\mathbf{x}) + \nabla\mathbf{f}(\mathbf{x})(\mathbf{y}_\gamma - \mathbf{x}) + \tfrac{\gamma^2}{2} \cdot \tfrac{2}{\gamma^2}\big[\mathbf{f}(\mathbf{y}_\gamma) - \mathbf{f}(\mathbf{x}) - \nabla f(\mathbf{x})(\mathbf{y}_\gamma - \mathbf{x})\big],\, \mathbf{y}_\gamma\big) \\
&\overset{(24)}{\leq} F\big(\mathbf{f}(\mathbf{x}) + \nabla\mathbf{f}(\mathbf{x})(\mathbf{y}_\gamma - \mathbf{x}),\, \mathbf{y}_\gamma\big) + \tfrac{\gamma^2}{2} F\big(\tfrac{2}{\gamma^2}\big[\mathbf{f}(\mathbf{y}_\gamma) - \mathbf{f}(\mathbf{x}) - \nabla f(\mathbf{x})(\mathbf{y}_\gamma - \mathbf{x})\big],\, \mathbf{y}_\gamma\big) \\
&\;\leq \; F\big(\mathbf{f}(\mathbf{x}) + \nabla\mathbf{f}(\mathbf{x})(\mathbf{y}_\gamma - \mathbf{x}),\, \mathbf{y}_\gamma\big) + \tfrac{\gamma^2}{2}\mathcal{S},
\end{aligned}$$

which is the desired bound. ■

A.1. Proof of Theorem 3.1

Theorem 3.1 *Let Assumptions 1, 2a, and 3 be satisfied. Let* $\gamma_k := \min\{1, \frac{\Delta_k}{\mathcal{S}}\}$ ***or*** $\gamma_k := \frac{2}{2+k}$. *Then, for* $k \geq 1$ *it holds that*

$$\varphi(\mathbf{y}_k) - \varphi^\star \;\; \leq \;\; \frac{2\mathcal{S}}{1+k} \qquad \text{and} \qquad \min_{1 \leq i \leq k} \Delta_i \;\; \leq \;\; \frac{6\mathcal{S}}{k}. \tag{16}$$

Proof Indeed, for one iteration of the method, we have

$$\begin{aligned}
\varphi(\mathbf{y}_{k+1}) \;\; &\overset{(23)}{\leq} \;\; F\big(\mathbf{f}(\mathbf{y}_k) + \nabla\mathbf{f}(\mathbf{y}_k)(\mathbf{y}_{k+1} - \mathbf{y}_k),\, \mathbf{y}_{k+1}\big) + \tfrac{\gamma_k^2}{2}\mathcal{S} \\
&= \;\; F\big((1-\gamma_k)\mathbf{f}(\mathbf{y}_k) + \gamma_k(\mathbf{f}(\mathbf{y}_k) + \nabla\mathbf{f}(\mathbf{y}_k)(\mathbf{x}_{k+1} - \mathbf{y}_k)), \\
&\qquad\quad (1-\gamma_k)\mathbf{y}_k + \gamma_k\mathbf{x}_{k+1}\big) + \tfrac{\gamma_k^2}{2}\mathcal{S} \\
&\overset{(*)}{\leq} \;\; \varphi(\mathbf{y}_k) + \gamma_k\Big[F\big(\mathbf{f}(\mathbf{y}_k) + \nabla\mathbf{f}(\mathbf{y}_k)(\mathbf{x}_{k+1} - \mathbf{y}_k), \mathbf{x}_{k+1}\big) - \varphi(\mathbf{y}_k)\Big] + \tfrac{\gamma_k^2}{2}\mathcal{S} \\
&\equiv \;\; \varphi(\mathbf{y}_k) - \gamma_k\Delta_k + \tfrac{\gamma_k^2}{2}\mathcal{S},
\end{aligned}$$

where we used in $(*)$ that $F(\cdot,\cdot)$ is jointly convex. Hence, we obtain the following inequality for the progress of one step, for $k \geq 0$:

$$\varphi(\mathbf{y}_k) - \varphi(\mathbf{y}_{k+1}) \quad \geq \quad \gamma_k \Delta_k - \tfrac{\gamma_k^2}{2} \mathcal{S}. \tag{25}$$

Now, let us choose use an auxiliary sequence $A_k := k \cdot (k+1)$ and $a_{k+1} := A_{k+1} - A_k = 2(k+1)$. Then,

$$\tfrac{a_{k+1}}{A_{k+1}} \quad = \quad \tfrac{2}{2+k},$$

which is one of the possible choices for γ_k. Note that for the other choice, we set $\gamma_k = \min\{1, \frac{\Delta_k}{\mathcal{S}}\}$, which maximizes the right hand side of (25) with respect to $\gamma_k \in [0,1]$. Hence, in both cases we have that

$$\varphi(\mathbf{y}_k) - \varphi(\mathbf{y}_{k+1}) \quad \geq \quad \tfrac{a_{k+1}}{A_{k+1}} \Delta_k - \tfrac{a_{k+1}^2}{2A_{k+1}^2} \mathcal{S}, \tag{26}$$

or, rearranging the terms,

$$\begin{aligned} A_{k+1}\big[\varphi(\mathbf{y}_{k+1}) - \varphi^\star\big] \quad &\overset{(26)}{\leq} \quad A_{k+1}\big[\varphi(\mathbf{y}_k) - \varphi^\star\big] - a_{k+1}\Delta_k + \tfrac{a_{k+1}^2}{2A_{k+1}} \mathcal{S} \\ &\overset{(15)}{\leq} \quad A_k\big[\varphi(\mathbf{y}_k) - \varphi^\star\big] + \tfrac{a_{k+1}^2}{2A_{k+1}} \mathcal{S}. \end{aligned}$$

Telescoping this bound for the first $k \geq 1$ iterations, we get

$$\varphi(\mathbf{y}_k) - \varphi^\star \quad \leq \quad \tfrac{\mathcal{S}}{2A_k} \cdot \sum_{i=1}^{k} \tfrac{a_i^2}{A_i} \quad = \quad \tfrac{2\mathcal{S}}{k(k+1)} \cdot \sum_{i=1}^{k} \tfrac{i}{i+1} \quad \leq \quad \tfrac{2\mathcal{S}}{k+1}. \tag{27}$$

It remains to prove the convergence in terms of the accuracy measure Δ_k. For that, we telescope the bound (26), which is

$$a_{k+1}\Delta_k \quad \leq \quad a_{k+1}\varphi(\mathbf{y}_k) + A_k\varphi(\mathbf{y}_k) - A_{k+1}\varphi(\mathbf{y}_{k+1}) + \tfrac{a_{k+1}^2}{A_{k+1}} \tfrac{\mathcal{S}}{2}, \tag{28}$$

for the $k \geq 1$ iterations, and use the convergence for the functional residual (27):

$$\begin{aligned} \sum_{i=1}^{k} a_{i+1} \cdot \min_{1 \leq i \leq k} \Delta_i \quad &\leq \quad \sum_{i=1}^{k} a_{i+1}\Delta_i \\ &\overset{(28)}{\leq} \quad a_1\big[\varphi(\mathbf{y}_1) - \varphi^\star\big] + \sum_{i=1}^{k} a_{i+1}\big[\varphi(\mathbf{y}_i) - \varphi^\star\big] + \tfrac{\mathcal{S}}{2} \sum_{i=1}^{k} \tfrac{a_{i+1}^2}{A_{i+1}} \\ &\overset{(27)}{\leq} \quad 2\mathcal{S} \cdot \Big(1 + \sum_{i=1}^{k} \tfrac{a_{i+1}}{i+1} + \sum_{i=1}^{k} \tfrac{i}{i+1}\Big) \quad \leq \quad 2\mathcal{S} \cdot (1 + 3k). \end{aligned}$$

To finish the proof, we need to divide both sides by $\sum\limits_{i=1}^{k} a_{i+1} = A_{k+1} - a_1 = k(3+k)$. ■

A.2. Proof of Theorem 3.2

Theorem 3.2 *Let Assumptions 1 and 2a be satisfied. Let $\gamma_k := \min\{1, \frac{\Delta_k}{\mathcal{S}}\}$ **or** $\gamma_k := \frac{1}{\sqrt{1+k}}$. Then, for all $k \geq 1$ it holds that*

$$\min_{0 \leq i \leq k} \Delta_i \quad \leq \quad \frac{\varphi(\mathbf{y}_0) - \varphi^\star + 0.5\mathcal{S}(1+\ln(k+1))}{\sqrt{k+1}}. \tag{17}$$

Proof As in the proof of the previous theorem, our main inequality (25) on the progress of one step is:

$$\varphi(\mathbf{y}_k) - \varphi(\mathbf{y}_{k+1}) \quad \geq \quad \gamma_k \Delta_k - \frac{\gamma_k^2}{2}\mathcal{S},$$

where we can substitute $\gamma_k = \frac{1}{\sqrt{k+1}}$ for the both strategies of choosing this parameter.

Summing up this bound for the first $k+1$ iterations, we obtain

$$\sum_{i=0}^{k} \gamma_i \Delta_i \quad \leq \quad \varphi(\mathbf{y}_0) - \varphi(\mathbf{y}_{k+1}) + \frac{\mathcal{S}}{2} \sum_{i=0}^{k} \gamma_i^2. \tag{29}$$

Using the bound $\varphi(\mathbf{y}_{k+1}) \geq \varphi^\star$ and our value of γ_i, we get

$$\begin{aligned} \min_{0 \leq i \leq k} \Delta_i \cdot \sqrt{k+1} \quad &\leq \quad \sum_{i=0}^{k} \frac{\Delta_i}{\sqrt{1+i}} \quad \overset{(29)}{\leq} \quad \varphi(\mathbf{y}_0) - \varphi^\star + \frac{\mathcal{S}}{2} \sum_{i=0}^{k} \frac{1}{1+i} \\ &\leq \quad \varphi(\mathbf{y}_0) - \varphi^\star + \frac{\mathcal{S}}{2}(1 + \ln(k+1)), \end{aligned}$$

which is (17). ■

A.3. Proof of Theorem 4.1

Theorem 4.1 *Let Assumptions 1, 2b, and 3 be satisfied. We choose $\gamma_k := \frac{3}{k+3}$, $\beta_k := cF(\mathbf{L})\gamma_k$ and $\eta_k := \frac{\delta}{3(k+1)(k+2)}$ where $\delta > 0$ and $c \geq 0$ are chosen constants, and $F(\mathbf{L}) := \sup_{\mathbf{x} \in \mathcal{X}} F(\mathbf{L}, \mathbf{x})$. Then, for all $k \geq 1$ it holds that*

$$\varphi(\mathbf{y}_k) - \varphi^\star \quad \leq \quad \frac{\delta + 8cF(\mathbf{L})\mathcal{D}_{\mathcal{X}}^2}{(k+2)(k+3)} + \frac{2\max\{0, 1-c\}F(\mathbf{L})\mathcal{D}_{\mathcal{X}}^2}{k+3}.$$

Proof Let us consider one iteration of the method, for some $k \geq 0$.

Since all the components of $\mathbf{f}$ have the Lipschitz continuous gradients, it hold that

$$\mathbf{f}(\mathbf{y}_{k+1}) \quad \leq \quad \mathbf{f}(\mathbf{z}_{k+1}) + \nabla \mathbf{f}(\mathbf{z}_{k+1})(\mathbf{y}_{k+1} - \mathbf{z}_{k+1}) + \frac{\mathbf{L}}{2}\|\mathbf{y}_{k+1} - \mathbf{z}_{k+1}\|^2,$$

where the vector inequality is component-wise. Then, using the properties of F, we have

$$\begin{aligned}
\varphi(\mathbf{y}_{k+1}) \quad &= \quad F(\mathbf{f}(\mathbf{y}_{k+1}), \mathbf{y}_{k+1}) \\
&\overset{(8),(24)}{\leq} \quad F(\mathbf{f}(\mathbf{z}_{k+1}) + \nabla \mathbf{f}(\mathbf{z}_{k+1})(\mathbf{y}_{k+1} - \mathbf{z}_{k+1}), \mathbf{y}_{k+1}) + \tfrac{F(\mathbf{L})}{2}\|\mathbf{y}_{k+1} - \mathbf{z}_{k+1}\|^2 \\
&= \quad F\big((1-\gamma_k)\big[\mathbf{f}(\mathbf{z}_{k+1}) + \nabla \mathbf{f}(\mathbf{z}_{k+1})(\mathbf{y}_k - \mathbf{z}_{k+1})\big] \\
&\qquad\qquad + \gamma_k\big[\mathbf{f}(\mathbf{z}_{k+1}) + \nabla \mathbf{f}(\mathbf{z}_{k+1})(\mathbf{x}_{k+1} - \mathbf{z}_{k+1})\big], \\
&\qquad\qquad\quad (1-\gamma_k)\mathbf{y}_k + \gamma_k \mathbf{x}_{k+1}\big) \;+\; \tfrac{\gamma_k^2 F(\mathbf{L})}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\
&\leq \quad (1-\gamma_k)F(\mathbf{f}(\mathbf{z}_{k+1}) + \nabla \mathbf{f}(\mathbf{z}_{k+1})(\mathbf{y}_k - \mathbf{z}_{k+1}),\, \mathbf{y}_k) \\
&\qquad + \gamma_k F(\mathbf{f}(\mathbf{z}_{k+1}) + \nabla \mathbf{f}(\mathbf{z}_{k+1})(\mathbf{x}_{k+1} - \mathbf{z}_{k+1}),\, \mathbf{x}_{k+1}) \;+\; \tfrac{\gamma_k^2 F(\mathbf{L})}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\
&\leq \quad (1-\gamma_k)\varphi(\mathbf{y}_k) \;+\; \gamma_k F(\mathbf{f}(\mathbf{z}_{k+1}) + \nabla \mathbf{f}(\mathbf{z}_{k+1})(\mathbf{x}_{k+1} - \mathbf{z}_{k+1}),\, \mathbf{x}_{k+1}) \\
&\qquad\qquad + \tfrac{\gamma_k^2 F(\mathbf{L})}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2,
\end{aligned}$$

where the second equality comes from the update rule of $\mathbf{y}_{k+1}$, the second inequality comes from the joint convexity in Assumption 1, the third inequality comes from convexity of the components of $\mathbf{f}$ and monotonicity of F.

Since we are introducing a norm-regularized minimization subproblem for the purpose of acceleration, the term $\frac{\gamma_k^2 F(\mathbf{L})}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2$ can be further upper bounded using η_k-approximate guarantee (20), as follows, for any $\mathbf{x} \in \mathcal{X}$:

$$\begin{aligned}
\tfrac{\gamma_k^2 F(\mathbf{L})}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \quad &= \quad \Big(\tfrac{\gamma_k^2 F(\mathbf{L})}{2} - \tfrac{\beta_k \gamma_k}{2}\Big)\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + \tfrac{\beta_k \gamma_k}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\
&\leq \quad \tfrac{\beta_k \gamma_k}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2^2 + \tfrac{\gamma_k^2 F(\mathbf{L})(1-c)}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\
&= \quad \tfrac{\beta_k \gamma_k}{2}\big(\|\mathbf{x} - \mathbf{x}_k\|_2^2 - \|\mathbf{x} - \mathbf{x}_{k+1}\|^2 - 2\langle \mathbf{x}_k - \mathbf{x}_{k+1}, \mathbf{x}_{k+1} - \mathbf{x}\rangle\big) \\
&\qquad + \tfrac{\gamma_k^2 F(\mathbf{L})(1-c)}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\
&\overset{(20)}{\leq} \quad \tfrac{\beta_k \gamma_k}{2}\big(\|\mathbf{x} - \mathbf{x}_k\|_2^2 - \|\mathbf{x} - \mathbf{x}_{k+1}\|^2\big) + \tfrac{\gamma_k^2 F(\mathbf{L})(1-c)}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\
&\qquad + \gamma_k F(\mathbf{f}(\mathbf{z}_{k+1}) + \nabla \mathbf{f}(\mathbf{z}_{k+1})(\mathbf{x} - \mathbf{z}_{k+1}), \mathbf{x}) \\
&\qquad - \gamma_k F(\mathbf{f}(\mathbf{z}_{k+1}) + \nabla \mathbf{f}(\mathbf{z}_{k+1})(\mathbf{x}_{k+1} - \mathbf{z}_{k+1}), \mathbf{x}_{k+1}) + \gamma_k \eta_k,
\end{aligned}$$

where we used our choice $\beta_k = cF(\mathbf{L})\gamma_k$.

Therefore, combining these two bounds together, we obtain

$$\begin{aligned}\varphi(\mathbf{y}_{k+1}) &\leq (1-\gamma_k)\varphi(\mathbf{y}_k) + \gamma_k F(\mathbf{f}(\mathbf{z}_{k+1}) + \nabla \mathbf{f}(\mathbf{z}_{k+1})(\mathbf{x}-\mathbf{z}_{k+1}), \mathbf{x}) \\ &\quad + \tfrac{\beta_k\gamma_k}{2}\left(\|\mathbf{x}-\mathbf{x}_k\|^2 - \|\mathbf{x}-\mathbf{x}_{k+1}\|^2\right) + \tfrac{\gamma_k^2 F(\mathbf{L})(1-c)}{2}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|^2 + \gamma_k\eta_k \\ &\leq (1-\gamma_k)\varphi(\mathbf{y}_k) + \gamma_k\varphi(\mathbf{x}) + \tfrac{\beta_k\gamma_k}{2}\left(\|\mathbf{x}-\mathbf{x}_k\|^2 - \|\mathbf{x}-\mathbf{x}_{k+1}\|^2\right) \\ &\quad + \tfrac{\gamma_k^2 F(\mathbf{L})(1-c)}{2}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|^2 + \gamma_k\eta_k,\end{aligned}$$

for all $\mathbf{x} \in \mathcal{X}$, where we used convexity of $\mathbf{f}$ and monotonicity of F.

We now subtract $\varphi(\mathbf{x})$ from both sides, let $\mathbf{x} = \mathbf{x}^\star$ and denote $\varepsilon_k := \varphi(\mathbf{y}_k) - \varphi^\star$, which gives

$$\begin{aligned}\varepsilon_{k+1} &\leq (1-\gamma_k)\varepsilon_k + \tfrac{\gamma_k\beta_k}{2}\left(\|\mathbf{x}_k-\mathbf{x}^\star\|^2 - \|\mathbf{x}_{k+1}-\mathbf{x}^\star\|^2\right) \\ &\quad + \tfrac{\gamma_k^2 F(\mathbf{L})(1-c)}{2}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|^2 + \gamma_k\eta_k.\end{aligned} \tag{30}$$

We now move on to choosing the parameters γ_k, η_k and β_k in a way that allows us to accelerate. For more flexibility, we let $\gamma_k := \frac{a_{k+1}}{A_{k+1}}$, for some sequences $\{a_k\}_{k\geq 0}$ and $\{A_k\}_{k\geq 0}$ that will be defined later. Then (30) becomes:

$$\begin{aligned}A_{k+1}\varepsilon_{k+1} &\leq A_0\varepsilon_0 + \sum_{i=0}^{k} a_{i+1}\eta_i + \tfrac{1}{2}\sum_{i=0}^{k} a_{i+1}\beta_i\left(\|\mathbf{x}_i-\mathbf{x}^\star\|^2 - \|\mathbf{x}_{i+1}-\mathbf{x}^\star\|^2\right) \\ &\quad + \tfrac{F(\mathbf{L})(1-c)}{2}\sum_{i=0}^{k}\tfrac{a_{i+1}^2}{A_{i+1}}\|\mathbf{x}_{i+1}-\mathbf{x}_i\|^2 \\ &\leq A_0\varepsilon_0 + \sum_{i=0}^{k} a_{i+1}\eta_i + \tfrac{a_1\beta_0}{2}\|\mathbf{x}_0-\mathbf{x}^\star\|^2 + \tfrac{1}{2}\textstyle\sum_{i=1}^{k}(a_{i+1}\beta_i - a_i\beta_{i-1})\|\mathbf{x}_i-\mathbf{x}^\star\|^2 \\ &\quad + \tfrac{F(\mathbf{L})(1-c)}{2}\sum_{i=0}^{k}\tfrac{a_{i+1}^2}{A_{i+1}}\|\mathbf{x}_{i+1}-\mathbf{x}_i\|^2\end{aligned}$$

and therefore we have

$$\begin{aligned}\varepsilon_{k+1} &\leq \tfrac{A_0\varepsilon_0}{A_{k+1}} + \tfrac{1}{A_{k+1}}\sum_{i=0}^{k} a_{i+1}\eta_i + \tfrac{a_1\beta_0}{2A_{k+1}}\|\mathbf{x}_0-\mathbf{x}^\star\|^2 \\ &\quad + \tfrac{1}{2A_{k+1}}\sum_{i=1}^{k}(a_{i+1}\beta_i - a_i\beta_{i-1})\|\mathbf{x}_i-\mathbf{x}^\star\|^2 + \tfrac{F(\mathbf{L})(1-c)}{2A_{k+1}}\sum_{i=0}^{k}\tfrac{a_{i+1}^2}{A_{i+1}}\|\mathbf{x}_{i+1}-\mathbf{x}_i\|^2.\end{aligned}$$

We wish to choose sequences A_k, a_k, β_k and η_k such that we obtain a $\mathcal{O}\left(1/k^2\right)$ rate of convergence on the functional residual of $\varphi(\cdot)$. The constraint we require on the sequences is $\gamma_k F(\mathbf{L}) \leq \beta_k$. The following choices

$$\begin{aligned}\eta_k &= \tfrac{\delta}{a_{k+1}}, \quad \text{for some constant } \delta > 0, \\ \beta_k &= cF(\mathbf{L})\gamma_k, \quad \text{for some constant } c > 0 \\ a_{k+1} &= A_{k+1} - A_k = \tfrac{3A_{k+1}}{k+3}, \quad A_{k+1} = (k+1)(k+2)(k+3),\end{aligned}$$

give us the desired outcome, since equation (31) becomes:

$$\begin{aligned}\varepsilon_{k+1} &\leq \frac{\delta}{(k+2)(k+3)} + \frac{3cF(\mathbf{L})\parallel \mathbf{x}_0 - \mathbf{x}^\star \parallel^2}{(k+1)(k+2)(k+3)} + \frac{5ckF(\mathbf{L})\mathcal{D}_{\mathcal{X}}^2}{(k+1)(k+2)(k+3)} \\ &+ \frac{9F(\mathbf{L})(1-c)}{2(k+1)(k+2)(k+3)} \sum_{i=0}^{k} (i+1)\|\mathbf{x}_{i+1} - \mathbf{x}_i\|^2 \\ &\leq \frac{\delta}{(k+2)(k+3)} + \frac{8cF(\mathbf{L})\mathcal{D}_{\mathcal{X}}^2}{(k+2)(k+3)} + \frac{2\max\{0, 1-c\}F(\mathbf{L})\mathcal{D}_{\mathcal{X}}^2}{k+3}\end{aligned}$$

since $a_{i+1}\beta_i - a_i\beta_{i-1} = \frac{9cF(\mathbf{L})\left(i^2+5i+4\right)}{i^2+5i+6} < 9cF(\mathbf{L})$ and $\parallel \mathbf{x}_i - \mathbf{x}^\star \parallel^2 \leq \mathcal{D}_{\mathcal{X}}^2$. ■

A.4. Proof of Theorem 5.1

Theorem 5.1 *Let Assumptions 1, 2b, and 3 be satisfied. Then, for all $t \geq 1$ it holds that*

$$P(\mathbf{u}_t) - P^\star \quad \leq \quad \tfrac{2\beta\mathcal{D}_{\mathcal{X}}^2}{t+1} \qquad \textit{and} \qquad \min_{1 \leq i \leq t} \Delta_t \quad \leq \quad \tfrac{6\beta\mathcal{D}_{\mathcal{X}}^2}{t}.$$

Consequently, Algorithm 3 returns an η-approximate solution according to condition (20) after at most $\mathcal{O}\left(\frac{\beta\mathcal{D}_{\mathcal{X}}^2}{\eta}\right)$ iterations.

Proof Let us introduce our subproblem, in a general form, that is

$$s^\star \quad = \quad \min_{\mathbf{u} \in \mathcal{X}} \Big\{ s(\mathbf{u}) \quad \overset{\text{def}}{=} \quad r(\mathbf{u}) + h(\mathbf{u}), \Big\} \tag{31}$$

where $r(\cdot)$ is a differentiable convex function, whose gradient is Lipschitz continuous with constant $\beta > 0$, and $h(\mathbf{u})$ is a general proper closed convex function, not necessarily differentiable.

In our case, for computing the inexact proximal step (19), we set

$$\begin{aligned} r(\mathbf{u}) \quad &:= \quad \tfrac{\beta}{2}\|\mathbf{u} - \mathbf{x}\|^2, \\ h(\mathbf{u}) \quad &:= \quad F(\mathbf{f}(\mathbf{z}) + \nabla f(\mathbf{z})(\mathbf{u} - \mathbf{z}), \mathbf{u}), \end{aligned}$$

for a fixed $\mathbf{x}$ and $\mathbf{z}$.

Then, in each iteration of Algorithm 3, we compute, for $t \geq 0$:

$$\mathbf{v}_{t+1} \quad \in \quad \operatorname*{Argmin}_{\mathbf{u} \in \mathcal{X}} \Big\{ \ \langle \nabla r(\mathbf{u}_t), \mathbf{u} \rangle + h(\mathbf{u}) \ \Big\}. \tag{32}$$

The optimality condition for this operation is (see, e.g. Theorem 3.1.23 in Nesterov (2018b))

$$\langle \nabla r(\mathbf{u}_t), \mathbf{u} - \mathbf{v}_{t+1} \rangle + h(\mathbf{u}) \quad \geq \quad h(\mathbf{v}_{t+1}), \qquad \forall \mathbf{u} \in \mathcal{X}. \tag{33}$$

Therefore, employing the Lipschitz continuity of the gradient of $r(\cdot)$, we have

$$\begin{aligned} s(\mathbf{u}_{t+1}) &\leq r(\mathbf{u}_t) + \langle \nabla r(\mathbf{u}_t), \mathbf{u}_{t+1} - \mathbf{u}_t \rangle + \tfrac{\beta}{2}\|\mathbf{u}_{t+1} - \mathbf{u}_t\|^2 + h(\mathbf{u}_{t+1}) \\ &= r(\mathbf{u}_t) + \alpha_t \langle \nabla r(\mathbf{u}_t), \mathbf{v}_{t+1} - \mathbf{u}_t \rangle + \tfrac{\beta\alpha_t^2}{2}\|\mathbf{v}_{t+1} - \mathbf{u}_t\|^2 \\ &\qquad + h(\alpha_t \mathbf{v}_{t+1} + (1-\alpha_t)\mathbf{u}_t) \qquad (34) \\ &\leq s(\mathbf{u}_t) + \alpha_t \big(\langle \nabla r(\mathbf{u}_t), \mathbf{v}_{t+1} - \mathbf{u}_t \rangle + h(\mathbf{v}_{t+1}) - h(\mathbf{u}_t) \big) + \tfrac{\beta\alpha^2\mathcal{D}_{\mathcal{X}}^2}{2} \\ &\equiv s(\mathbf{u}_t) - \alpha_t \Delta_t + \tfrac{\beta\alpha_t^2\mathcal{D}_{\mathcal{X}}^2}{2}, \end{aligned}$$

where the last equality comes from the definition of Δ_t in Algorithm 3.

Note that α_t is defined as the minimizer of $\frac{\beta\alpha_t^2}{2}\|\mathbf{v}_{t+1} - \mathbf{u}_t\|^2 - \alpha_t\Delta_t$ and hence, for any other $\rho_t \in [0,1]$ it will hold that:

$$s(\mathbf{u}_{t+1}) \leq s(\mathbf{u}_t) - \rho_t\Delta_t + \frac{\beta\rho_t^2\mathcal{D}_{\mathcal{X}}^2}{2}. \qquad (35)$$

At the same time,

$$\begin{aligned} \Delta_t &\overset{\text{def}}{=} h(\mathbf{u}_t) - h(\mathbf{v}_{t+1}) - \langle \nabla r(\mathbf{u}_t), \mathbf{v}_{t+1} - \mathbf{u}_t \rangle \\ &\overset{(32)}{\geq} h(\mathbf{u}_t) - h(\mathbf{u}) - \langle \nabla r(\mathbf{u}_t), \mathbf{u} - \mathbf{u}_t \rangle \qquad (36) \\ &\geq s(\mathbf{u}_t) - s(\mathbf{u}), \qquad \forall \mathbf{u} \in \mathcal{X} \end{aligned}$$

where the last line follows from the convexity of $r(\mathbf{u})$. Letting $\mathbf{u} := \mathbf{u}^\star$ (solution to (31)) in (36) and further substituting it into (35) and subtracting $\mathbf{s}^\star$ from both sides, we obtain

$$[s(\mathbf{u}_{t+1}) - s^\star] \leq (1-\rho_t)\,[s(\mathbf{u}_t) - \mathbf{s}^\star] + \tfrac{\beta\rho_t^2\mathcal{D}_{\mathcal{X}}^2}{2}. \qquad (37)$$

Now, let us choose $\rho_t := \frac{a_{t+1}}{A_{t+1}}$ for sequences $A_t := t\cdot(t+1)$, and $a_{t+1} := A_{t+1} - A_t = 2(t+1)$. Then, $\rho_t := \frac{2}{2+t}$, $t \geq 0$. Using this choice, inequality (37) can be rewritten as

$$A_{t+1}\big[s(\mathbf{u}_{t+1}) - s^\star\big] \leq A_t\big[s(\mathbf{u}_t) - s^\star\big] + \tfrac{a_{t+1}^2\beta\mathcal{D}_{\mathcal{X}}^2}{2A_{t+1}}$$

Telescoping this inequality for the first iterations, we obtain, for $t \geq 1$:

$$s(\mathbf{u}_t) - s^\star \leq \tfrac{\beta\mathcal{D}_{\mathcal{X}}^2}{2A_t} \cdot \sum_{i=1}^{t} \tfrac{a_i^2}{A_i} = \tfrac{\beta\mathcal{D}_{\mathcal{X}}^2}{2t(t+1)} \cdot \sum_{i=1}^{t} \tfrac{4i}{i+1} \leq \tfrac{2\beta\mathcal{D}_{\mathcal{X}}^2}{t+1}. \qquad (38)$$

This is the global convergence in terms of the functional residual. It remains to justify the convergence for the accuracy certificates Δ_t. Multiplying (35) by A_{t+1}, we obtain

$$a_{t+1}\Delta_t \leq a_{t+1}s(\mathbf{u}_t) + A_t s(\mathbf{u}_t) - A_{t+1}s(\mathbf{u}_{t+1}) + \tfrac{a_{t+1}^2}{A_{t+1}}\tfrac{\beta\mathcal{D}_{\mathcal{X}}^2}{2}. \qquad (39)$$

Telescoping this bound, we get, for $t \geq 1$:

$$\begin{array}{rcl}
\sum\limits_{i=1}^{t} a_{i+1} \cdot \min\limits_{1 \leq i \leq t} \Delta_i & \leq & \sum\limits_{i=1}^{t} a_{i+1} \Delta_i \\
& \overset{(39)}{\leq} & a_1 \big[s(\mathbf{u}_1) - s^\star \big] + \sum\limits_{i=1}^{t} a_{i+1} \big[s(\mathbf{u}_i) - s^\star \big] + \frac{\beta \mathcal{D}_{\mathcal{X}}^2}{2} \sum\limits_{i=1}^{t} \frac{a_{i+1}^2}{A_{i+1}} \\
& \overset{(38)}{\leq} & 2\beta \mathcal{D}_{\mathcal{X}}^2 \cdot \Big(1 + \sum\limits_{i=1}^{t} \frac{a_{i+1}}{i+1} + \frac{1}{4} \sum\limits_{i=1}^{t} \frac{a_{i+1}^2}{A_{i+1}} \Big) \\
& \leq & 2\beta \mathcal{D}_{\mathcal{X}}^2 \cdot (1 + 3t).
\end{array}$$

Dividing both sides by $\sum_{i=1}^{t} a_{i+1} = A_{t+1} - A_1 = t(3+t)$ completes the proof we finally get:

$$\min_{1 \leq i \leq t} \Delta_i \quad \leq \quad \frac{6\beta \mathcal{D}_{\mathcal{X}}^2}{t}.$$

■

A.5. Proof of Proposition 3.1

Proposition 3.1 *Let $\gamma_k := \frac{1}{\sqrt{1+k}}$. Then, for the iterations (18), under Assumption 2b and for all $k \geq 1$, it holds that*

$$\min_{0 \leq i \leq k} \max_{\mathbf{y} \in \mathcal{X}} \langle \nabla \Phi(\mathbf{y}_i), \mathbf{y}_i - \mathbf{y} \rangle \quad \leq \quad \mathcal{O}\big(\tfrac{\ln(k)}{\sqrt{k}} \big).$$

Proof In our case, we have $\varphi(\mathbf{x}) \equiv \|\mathbf{f}(\mathbf{x})\|_2$. Using Lemma A.1, we obtain

$$\begin{array}{rcl}
\varphi(\mathbf{y}_{k+1}) & \leq & \|\mathbf{f}(\mathbf{y}_k) + \nabla f(\mathbf{y}_k)(\mathbf{y}_{k+1} - \mathbf{y}_k)\|_2 + \frac{\gamma_k^2}{2} \mathcal{S} \\
& = & \|\mathbf{f}(\mathbf{y}_k) + \gamma_k \nabla f(\mathbf{y}_k)(\mathbf{x}_{k+1} - \mathbf{y}_k)\|_2 + \frac{\gamma_k^2}{2} \mathcal{S},
\end{array} \tag{40}$$

where $\mathbf{x}_{k+1} \in \mathcal{X}$ is the point such that $\mathbf{y}_{k+1} = \mathbf{y}_k + \gamma_k (\mathbf{x}_{k+1} - \mathbf{y}_k)$. Using convexity of the function $g(\mathbf{x}) \overset{\text{def}}{=} \|\mathbf{f}(\mathbf{y}_k) + \gamma_k \nabla \mathbf{f}(\mathbf{y}_k)(\mathbf{x} - \mathbf{y}_k)\|_2$, we get that

$$\begin{array}{rcl}
\varphi(\mathbf{y}_k) \;\; = \;\; g(\mathbf{y}_k) & \geq & g(\mathbf{x}_{k+1}) + \langle g'(\mathbf{x}_{k+1}), \mathbf{y}_k - \mathbf{x}_{k+1} \rangle \\
& = & \|\mathbf{f}(\mathbf{y}_k) + \gamma_k \nabla f(\mathbf{y}_k)(\mathbf{x}_{k+1} - \mathbf{y}_k)\|_2 + \langle g'(\mathbf{x}_{k+1}), \mathbf{y}_k - \mathbf{x}_{k+1} \rangle,
\end{array}$$

where the subgradient $g'(\mathbf{x}_{k+1}) = \gamma_k \nabla \mathbf{f}(\mathbf{y}_k)^\top \frac{\mathbf{f}_{k+1}}{\|\mathbf{f}_{k+1}\|_2}$ with $\mathbf{f}_{k+1} \overset{\text{def}}{=} \mathbf{f}(\mathbf{y}_k) + \gamma_k \nabla \mathbf{f}(\mathbf{y}_k)(\mathbf{x}_{k+1} - \mathbf{y}_k)$, satisfies the stationary condition for the method step:

$$\langle g'(\mathbf{x}_{k+1}), \mathbf{x} - \mathbf{x}_{k+1} \rangle \quad \geq \quad 0, \qquad \forall \mathbf{x} \in \mathcal{X}. \tag{41}$$

A few comments are in order now about the use of the subgradient above. Note that we wish to impose an assumption on $\mathbf{f}$ which can ensure that $\mathbf{f}(\mathbf{y}_k) + \gamma_k \nabla \mathbf{f}(\mathbf{y}_k)(\mathbf{x} - \mathbf{y}_k) \neq \mathbf{0} \in \mathbb{R}^n$. First, some preliminaries. Under Assumption 2b on $\mathbf{f}$, it holds that:

$$\exists \mathcal{F} \in (0, \infty) \text{ s.t. } \|\, \mathbf{f}(x) \,\| \leq \mathcal{F}, \forall \mathbf{x} \in \mathcal{X} \quad \text{by continuity of } \mathbf{f} \tag{42}$$

$$\exists \mathcal{G} \in (0, \infty) \text{ s.t. } \|\, \nabla \mathbf{f}(x) \,\| \leq \mathcal{G}, \forall \mathbf{x} \in \mathcal{X} \quad \text{by continuous differentiability of } \mathbf{f} \tag{43}$$

From here, we can bound the products between Jacobians and iterates as follows:

$$\|\, \nabla \mathbf{f}(\mathbf{x})(\mathbf{y} - \mathbf{z}) \,\| \leq \|\, \nabla \mathbf{f}(\mathbf{x}) \,\| \,\|\, \mathbf{y} - \mathbf{z} \,\| \leq \mathcal{G}\mathcal{D}_{\mathcal{X}}, \quad \forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{X}. \tag{44}$$

Thus, without loss of generality, we can shift $\mathbf{f}$ by a constant vector of identical values depending on $\mathcal{G}\mathcal{D}_{\mathcal{X}}$ such that we ensure, for example, $\mathbf{f}(\mathbf{y}_k) + \gamma_k \nabla \mathbf{f}(\mathbf{y}_k)(\mathbf{x} - \mathbf{y}_k) > \mathbf{0}$ component-wise. Hence, combining these observations with (40), we have

$$\begin{aligned} \varphi(\mathbf{y}_k) - \varphi(\mathbf{y}_{k+1}) \quad &\geq \quad \langle g'(\mathbf{x}_{k+1}), \mathbf{y}_k - \mathbf{x}_{k+1} \rangle - \tfrac{\gamma_k^2}{2} \mathcal{S} \\ &\overset{(41)}{\geq} \quad \max_{\mathbf{x} \in \mathcal{X}} \langle g'(\mathbf{x}_{k+1}), \mathbf{y}_k - \mathbf{x} \rangle - \tfrac{\gamma_k^2}{2} \mathcal{S}. \end{aligned}$$

Then, by lower bounding appropriately using (42) and (43), we get:

$$\begin{aligned} \varphi(\mathbf{y}_k) - \varphi(\mathbf{y}_{k+1}) \quad &\geq \quad \tfrac{\gamma_k}{\mathcal{F} + \mathcal{G}\mathcal{D}_{\mathcal{X}}} \max_{\mathbf{y} \in \mathcal{X}} \langle \nabla \mathbf{f}(\mathbf{y}_k)^\top \mathbf{f}(\mathbf{y}_k), \mathbf{y}_k - \mathbf{y} \rangle - \gamma_k^2 \Big(\tfrac{\mathcal{G}\mathcal{D}_{\mathcal{X}}^2}{\mathcal{F} + \mathcal{G}\mathcal{D}_{\mathcal{X}}} + \tfrac{\mathcal{S}}{2} \Big) \\ &= \quad \tfrac{\gamma_k}{\mathcal{F} + \mathcal{G}\mathcal{D}_{\mathcal{X}}} \max_{\mathbf{y} \in \mathcal{X}} \langle \nabla \Phi(\mathbf{y}_k), \mathbf{y}_k - \mathbf{y} \rangle - \gamma_k^2 \Big(\tfrac{\mathcal{G}\mathcal{D}_{\mathcal{X}}^2}{\mathcal{F} + \mathcal{G}\mathcal{D}_{\mathcal{X}}} + \tfrac{\mathcal{S}}{2} \Big). \end{aligned}$$

Substituting $\gamma_k := \frac{1}{\sqrt{1+k}}$ and telescoping this bound would lead to the desired global convergence (for the details, see the end of the proof of Theorem 3.2). ■

Appendix B. Interpretation of Δ_k in the non-convex setting

While we cannot make any strong claims about the meaning of Δ_k in general, we can provide an additional observation for this quantity when the outer component F is smooth inside a ball included in $\mathcal{X}$.

Thus, consider a ball of radius ε centered at $\mathbf{y}_k$ denoted by $B(\mathbf{y}_k, \varepsilon) = \{\mathbf{x} \in \mathbb{R}^d \ : \ \|\mathbf{x} - \mathbf{y}_k\| \leq \varepsilon\}$, and set $\mathcal{B} = B(\mathbf{y}_k, \varepsilon) \cap \mathcal{X}$. Assuming that $F(\mathbf{u}, \mathbf{x})$ is differentiable at all points from $\mathbb{R}^n \times \mathcal{B}$, and that its gradient is Lipschitz continuous with constant L_F, we have for any $\mathbf{x} \in \mathcal{B} \subseteq \mathcal{X}$:

$$\begin{aligned} \Delta_k \quad &= \quad \max_{\mathbf{x} \in \mathcal{X}} \Big[\varphi(\mathbf{y}_k) - F\big(\mathbf{f}(\mathbf{y}_k) + \nabla \mathbf{f}(\mathbf{y}_k)(\mathbf{x} - \mathbf{y}_k), \mathbf{x}\big) \Big] \\ &\geq \quad \max_{\mathbf{x} \in \mathcal{B}} \Big[\varphi(\mathbf{y}_k) - F(\mathbf{f}(\mathbf{y}_k), \mathbf{y}_k) - \langle \tfrac{\partial F}{\partial \mathbf{u}}(\mathbf{f}(\mathbf{y}_k), \mathbf{y}_k), \nabla \mathbf{f}(\mathbf{y}_k)(\mathbf{x} - \mathbf{y}_k) \rangle \\ &\qquad\qquad - \langle \tfrac{\partial F}{\partial \mathbf{x}}(\mathbf{f}(\mathbf{y}_k), \mathbf{y}_k), \mathbf{x} - \mathbf{y}_k \rangle - \tfrac{L_F}{2} \big(\|\nabla \mathbf{f}(\mathbf{y}_k)\|^2 + 1 \big) \cdot \varepsilon^2 \Big] \\ &= \quad \max_{\mathbf{x} \in \mathcal{B}} \Big[\langle \nabla \varphi(\mathbf{y}_k), \mathbf{y}_k - \mathbf{x} \rangle \Big] - \tfrac{L_F}{2} \big(\|\nabla \mathbf{f}(\mathbf{y}_k)\|^2 + 1 \big) \cdot \varepsilon^2. \end{aligned}$$

Hence, for a small enough ball, Δ_k is an $\mathcal{O}\left(\varepsilon^2\right)$-approximation of the original FW gap restricted to the considered neighborhood. If, in addition, the composite function φ is convex in $\mathcal{B}$ and there is a local optimum $\mathbf{x}^\star \in \mathcal{B}$, then Δ_k is an $\mathcal{O}\left(\varepsilon^2\right)$-approximation of functional suboptimality.

Appendix C. Additional Application Examples

Example 3 *We define a generalized nonlinear model as,*

$$F(\mathbf{u}, \mathbf{x}) \equiv \sum_{i=1}^{n} \phi(u^{(i)}), \tag{45}$$

where $\phi : \mathbb{R} \to \mathbb{R}$ is a fixed convex loss function, and n is the number of data points. Problem (4) then reduces to training a (non-convex) model, for example a neural network, with respect to the constraint set $\mathcal{X}$: $\min\limits_{\mathbf{x}\in\mathcal{X}} \sum\limits_{i=1}^{m} \phi(f_i(\mathbf{x}))$.

Solving this problem then involves training a linear model within the basic subroutine (5) $\min\limits_{\mathbf{x}\in\mathcal{X}} \sum\limits_{i=1}^{m} \phi(\langle \mathbf{a}_i, \mathbf{x}\rangle + b_i)$, which is a convex problem. Amongst the loss functions relevant to Machine Learning, the following are convex and subhomogeneous thus making F in (45) satisfy Assumption 1:

- *ℓ_1-regression: $\phi(t) = |t|$*
- *Hinge loss (SVM): $\phi(t) = \max\{0, t\}$*
- *Logistic loss: $\phi(t) = \log(1 + e^t)$*