Structure and complexity in non-convex and non-smooth optimization

Courtney Paquette

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2017

Reading Committee:

Dmitriy Drusvyatskiy, Chair

Jim Burke

John Sylvester

Program Authorized to Offer Degree:
Mathematics

University of Washington

Abstract

Structure and complexity in non-convex and non-smooth optimization

Courtney Paquette

Chair of the Supervisory Committee:
Assistant Professor Dmitriy Drusvyatskiy
Mathematics

Complexity theory drives much of modern optimization, allowing a fair comparison between competing numerical methods. The subject broadly seeks to both develop efficient algorithms and establish limitations on efficiencies of any algorithm for the problem class. Classical complexity theory based on oracle models targets problems that are both smooth and convex. Without smoothness, methods rely on exploiting the structure of the target function to improve on the worst-case complexity of non-smooth convex optimization. This thesis explores complexity of first-order methods for structured non-smooth and non-convex problems. A central example is the minimization of a composition of a convex function with a smooth map – the so-called convex-composite problem class. Nonlinear least squares formulations in engineering and nonlinear model fitting in statistics fall within this framework. The thesis develops new algorithms for the composite problem class, along with inertial variants that are adaptive to convexity.

Acceleration is a widely used term in contemporary optimization. The term is often used to describe methods with efficiency guarantees matching the best possible complexity estimates for a given problem class. This thesis develops methods that interpolate between convex and non-convex settings. In particular, we focus on minimizing large finite sum problems, popular for modeling empirical risk in statistical applications, when the user is unaware of the convexity of the objective function. The scheme we describe has convergence guarantees that adapt to the underlying convexity of the objective function.

First-order algorithms for non-smooth problems depend on having access to generalized derivatives of the objective function. We conclude the thesis with a fresh look at variational properties of spectral function. These are the functions on the space of symmetric matrices that depend on the matrix only through its eigenvalues. In particular, our analysis dramatically simplifies currently available derivations of differential formulas of such functions.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

This thesis was largely written over a two month period in early 2017, starting from previously completed collaborative work. Once I finally began writing it, I found – much to my surprise – that the thesis almost wrote itself; the lessons and experiences of the past five years at UW and before were enough to propel me through to the finish.

Hart Smith taught my first real analysis course and subsequent topics courses in PDE and harmonic analysis. I started with such a poor understanding of analysis- my proofs being too imprecise. With his guidance and support, he taught me to appreciate the details- to get my hands dirty with proofs, and value the "messiness" of mathematics. Beyond his mathematical prowess, he has been a wonderful friend; I will miss our walks around the UW campus.

Although I have only known Zaid Harchaoui for a year, he has become one of the most supportive advisors in my academic life. Zaid has been my introduction to the academic community, arranging for me to meet with new collaborates and present my ideas at conferences. His encouragement and guidance has been unparalleled.

Jacob Richey has been a constant source of inspiration to me. I could not have asked for a more welcoming and entertaining officemate. More than anyone I know, he's packed with incredible conjectures about the world that are instantaneously recognizable as his, and they inspire and impel me to more and greater things.

I met Jim Morrow in my senior year as an undergraduate. If it were not for him, I would not be here getting my Ph.D. in mathematics. He has believed in me from the beginning, unlike no one else. He is a constant source of wisdom and energy, advising me in a way that only he can.

Finally, I would be remiss if I didn't thank Dima, my adviser. I'm pretty certain there is nothing I can write here to adequately do justice to my relationship with Dima. Dima suggested to me my first project, and that turned into a great odyssey. I do not know if there is another person in the world who would come in on weekends or stay late to work out complicated bounds on your whiteboard. I'm all the wiser and all the smarter for having you as my advisor, and I would do it all again.

There are also people whose effect on me is profound and ineffable, but without whom I could not be where I am today: my mother, my husband Elliot, and a long list of friends here in Seattle and in places all around the world.

DEDICATION

To my family and friends.

Chapter 1

INTRODUCTION

Optimization is ubiquitous in our society. Notable advances emerged during WWII, the beginning of modern optimization, and these breakthroughs, such as the simplex method for linear programming, sparked a movement in the field which continues to flourish today. Modern optimization methods are routinely used in engineering, machine learning, and high dimensional statistics. Continuous optimization, a major theme in this thesis, focuses on the setting where the decision variables take values in $\mathbb{R}^n$ as opposed to a discrete set. Modern applications of continuous optimization seek to extract conclusions from immense data without sacrificing efficiency and accuracy to the minima – a formidable task for optimization specialists. Under these circumstances, the high per-iteration cost of computing second-order information (*e.g.* Hessian) is rendered forbidden. Instead, first-order methods, algorithms which rely purely on gradient and function value information, have dominated large-scale computing of late.

The simplest first-order method for minimizing a smooth function is the *gradient-descent scheme* and its variants. This scheme dates back to Cauchy [22] and is still widely used today, requiring at most $O(\varepsilon^{-2})$ number of iterations to obtain an ε-stationary point – a point satisfying $\|\nabla f(x)\| \leq \varepsilon$. If the target function is also convex, then the iterates produced by gradient descent converge to a *global minimizer*, and the above rate to stationarity automatically improves to $O(\varepsilon^{-1})$. Moreover, gradient descent ensures $f(x) - \inf f \leq \varepsilon$ after at most $O(\varepsilon^{-1})$ iterations. Naturally, one may question whether, for this function class, gradient descent gives the best possible convergence rate. Aiming to address questions of this type, Nemirovski and Yudin [78] developed a comprehensive complexity theory for convex minimization based on worst-case oracle models. The theory provides *lower complexity bounds* which stipulate that any "algorithm" needs at *least* a certain amount of "information" about the objective function to find an approximate solution; methods achieving such best possible efficiency estimates are called optimal for the problem class. In the early 1980's, Nesterov [81, 82] gave the first true first-order optimal algorithm for smooth convex minimization. The latter algorithm, known as *Nesterov's accelerated gradient method*, is one of today's most influential schemes for smooth optimization and achieves a complexity guarantee of $O(\varepsilon^{-1/2})$ in function sub-optimality.

Many large-scale applications require minimizing a sum of a smooth function and a "simple" convex function. Problems of this type are known in the literature as *additive composite minimization*. This, in essence, is the easiest type of non-smooth optimization problem for first-order methods. In particular, gradient descent and its accelerated variants (when the smooth function is also convex) naturally extend to this setting, with analogous complexity guarantees [4, 86]. Over the past five years, the additive composite class has been refined to a large finite sum-structure: $\min_x \ \frac{1}{m}\sum_{i=1}^m c_i(x) + g(x)$, where $c_i(x)$ are smooth and $g(x)$ is simple and convex. This type of optimization problem naturally arises when modelling empirical risk in statistical applications. The evaluation of the gradient of the sum $\frac{1}{n}\sum_{i=1}^m c_i(x)$ can not be viewed as a single unit of cost, but requires a pass through all the individual functions c_i. Methods, stochastic perhaps, that require fewer gradient evaluations of the individual functions c_i might be superior. Notable examples include stochastic gradient and dual averaging [84, 118] and incremental algorithms [34, 55, 99]. Settings where the c_i are non-convex, in particular, have been gaining much interest, in large part motivated by robust fidelity measures and deep learning. Chapter 3 is devoted to tackling the large non-convex finite-sum problem, developing a first-order "inertial meta-algorithm": a procedure for taking methods originally designed for the convex finite-sum problems and extending them so that they apply when the objective is non-convex.

More powerful modelling frameworks require a more sophisticated class of non-smooth and non-convex optimization problems than those that are additive composite. Much of the thesis (Chapter 2) concerns the *composite problem class*: $\min_x \ h(c(x)) + g(x)$, where h

is finite-valued and convex, g is merely convex, and c is a smooth map. This formulation encompasses a wide variety of applications, including nonlinear least squares in engineering, exact penalty formulations in nonlinear programming, nonlinear models in statistics, nonlinear Kalman filters, etc. The problem in question is both non-smooth and non-convex. Our work in this area is devoted to understanding the behavior of "critical points" of the composite problem class through the rich language of variational analysis [98] and developing numerical methods with provable efficiency guarantees. In particular, we show that there is a first-order method for this problem class that achieves the efficiency estimate $\widetilde{O}(\varepsilon^{-3})$ in an appropriate stationarity measure generalizing the gradient. Along the way, we explore the meaning of "accelerating" a first-order method for non-convex problems.

In concrete non-smooth and non-convex optimization problems, such as the convex composite problem class above, numerical methods rely on having access to generalized derivatives of the objective function. Chapter 4 in the thesis revisits this computation for the wide class of *spectral functions*; these are functions on the space of symmetric matrices that depend on the matrices only through their eigenvalues. The spectral norm and the $k'th$ largest eigenvalue are good examples.

The organization of the thesis is as follows. The rest of Chapter 1 is devoted to a more detailed discussion of the background material. Chapter 2 discusses the convex composite problem class, focusing on efficient first-order methods and "acceleration". Chapter 3 considers the idea of inertial acceleration more broadly, developing a generic inertial acceleration scheme for minimizing non-convex finite-sum problems. The final Chapter 4 revisits non-smooth behavior of eigenvalue functions, drastically simplifying previously available analysis. Throughout, the thesis emphasizes the interplay between complexity, conditioning, and use of structure in non-smooth optimization. The main chapters 2, 3, and 4 of the thesis contain material from the papers:

- D. Drusvyatskiy and C. Paquette. Efficiency of minimizing compositions of convex functions and smooth maps. *Preprint arXiv:1605.00125*, 2016

- C. Paquette, H. Lin, D. Drusvyatskiy, J. Mairal, and Z. Harchaoui. Catalyst acceleration for gradient-based non-convex optimization. *Preprint arXiv:1703.10993*, 2017

- D. Drusvyatskiy and C. Paquette. Variational analysis of spectral functions simplified. *Journal of Convex Analysis (to appear)*, 2016

1.1 Generalizing the derivative

Variational analysis generalizes the concepts of differential analysis to functions that are not differentiable. Much analysis of smooth functions relies on linear and quadratic approximation arising from derivatives; so it is natural to extend these notions to non-smooth functions as well. A principle goal of non-smooth optimization is the search for critical points of non-smooth functions, or points where the "derivative" is zero.

Throughout the rest of this thesis, we consider functions defined on a Euclidean space, denoted by $\mathbb{R}^n$, with an induced inner product $\langle \cdot, \cdot \rangle$. As motivation, we begin with the classical and familiar Gâteaux derivative for a function $f : \mathbb{R}^n \to \mathbb{R}$ at x, namely a linear functional $f'(x)$ satisfying

$$\lim_{t \to 0} \frac{f(x+tv) - f(x)}{t} - \langle f'(x), v \rangle = 0.$$

This traditional definition introduces the derivative through pairs of points lying in the graph of the function. In optimization, we are interested in one-sided limits; therefore it is more instructive to consider the geometry of the epigraph – the set of points lying above the graph. Fix a point $\bar{c}$ in a subset $C \subset \mathbb{R}^n$ and consider all the points x whose projection onto C is $\bar{c}$. The set of vectors $x - \bar{c}$ and their non-negative multiples forms the *proximal normal cone* to the set C at $\bar{c}$, and is denoted by $N_C^P(\bar{c})$. This is illustrated in Figure 1.1.

There is a natural extension of the normal cone of a set to functions $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ by considering the function's *epigraph*,

$$\text{epi}(f) = \{(x, r) \in \mathbb{R}^n \times \mathbb{R} \, : \, f(x) \le r\}.$$

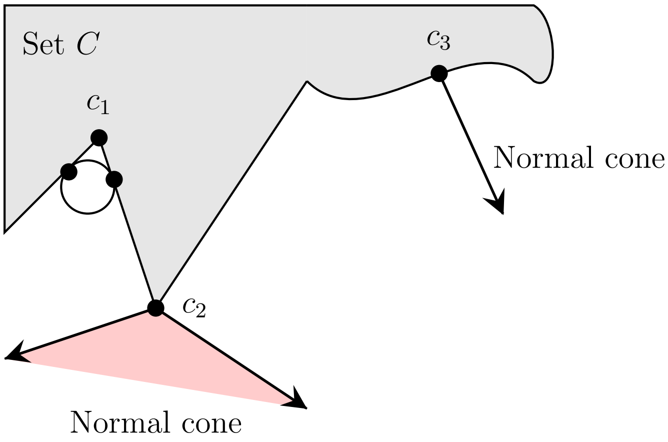


Figure 1.1: Depiction of proximal normal cones for a set C

This yields a "geometric" interpretation of a gradient. Suppose $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is a lower-semicontinuous function and x lies in $\operatorname{dom}(f)$. The *proximal subdifferential*, denoted by $\partial_P f(x)$, is the set of all vectors v such that $(v, -1) \in N_{\operatorname{epi}(f)}(x)$ and any element v of the proximal subdifferential is called a *proximal subgradient*. For example, the proximal subdifferential for the function $f(x) = |x|$ at $x = 0$ is $\partial_P f(0) = [-1, 1]$. An equivalent definition emulates the notion of local quadratic support for a function: a vector v lies in $\partial_P f(x)$ whenever there exist $\sigma, \eta > 0$ such that

$$f(y) \geq f(x) + \langle v, y - x \rangle - \sigma \|y - x\|^2 \quad \text{for all } y \in B(x, \eta)$$

where $B(x, \eta)$ denotes the ball centered at x with radius η.

A noted difference between the classical derivative and the proximal subdifferential is that it is not unique and may not exist for some points in the $\operatorname{dom}(f)$ as seen in the example $f(x) = -|x|$ at $x = 0$. When the function is C^2-smooth, the proximal subdifferential is unique and agrees with the classical Gâteaux derivative. Additional characterizations can be found in [98, Theorem 8.46].

An important deficit of the proximal subdifferential is that nearby subgradients can behave wildly differently. Since continuity of the derivative is generally desired in analysis, the analogous statement leads to the definition of the limiting subdifferential, denoted by $\partial f(x)$. A subgradient v is in the *limiting subdifferential* $\partial f(x)$ if there exist sequences x_i and $v_i \in \partial_P f(x_i)$ satisfying $(x_i, f(x_i), v_i) \to (x, f(x), v)$.

A main factor for constructing the theory of subdifferentials grew from the necessity to optimize non-smooth functions. However, guarantees of convergence to global minima require additional restrictions on the function class, namely *convexity.* A function $f : C \to \mathbb{R} \cup \{\infty\}$ is *convex* if the set C is convex and for any $x, y \in C$ and $\lambda \in [0, 1]$, it holds

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y).$$

For convex functions all the definitions of subdifferentials agree (see [9, Theorem 6.2.2], [24]) and in this case, we simply say the subdifferential of f and denote it by $\partial f(x)$. In particular, the subdifferential of a convex function reduces to

$$\partial f(x) = \{v \, : \, f(y) \geq f(x) + \langle v, y - x\rangle\}.$$

This lower linear approximation of the function drives complexity bounds for many optimization algorithms.

The role of convex functions in optimization is paramount. Convex functions naturally possess a *local to global* behavior. For a general C^1-smooth function, the linear approximation via the gradient only contains local behavior of the function, but with convex functions this linear approximation is a global lower bound. In particular, for convex functions, local minima are global minima:

Proposition 1.1.1 (Local minima are global minima)**.** *Suppose f is a convex function. Then the following are equivalent:*

(1) the point x^* is a local minima,

(2) the point x^* is a global minima,

(3) $0 \in \partial f(x^*)$.

From a numerical perspective, convexity improves convergence rates for algorithms.

1.1.1 Spectral Functions

A principle goal of non-smooth optimization is to find a "critical" point of a target function. A point x is *critical* (or *stationary*) for a function f if the inclusion $0 \in \partial f(x)$ holds; thus being able to compute the subdifferential of a function f is crucial. In Chapter 4, we explore functions defined on the space of symmetric $n \times n$ matrices that are orthogonally invariant i.e.

$$F(UXU^T) = F(X) \quad \text{for all } U \text{ orthogonal.}$$

These functions are often called *spectral* because they only depend on the matrix through its eigenvalues. Spectral functions naturally arise in the sciences and nowadays they appear throughout optimization and matrix analysis.

Each spectral function decomposes as $F(X) = (f \circ \lambda)(X)$, where λ maps to an *ordered* list of eigenvalues and f is symmetric (i.e. f is invariant under permutation of coordinates). Due to this correspondence with f, it is often easier to work directly with the symmetric function. Here are some common examples of symmetric and spectral functions:

- The nuclear norm $F(X) = \|X\|_*$ where $f(x) = \|x\|_1$
- $F(X) = \det(X)$ and $f(x) = \prod_{i=1}^n x_i$
- The Frobenius norm $F(X) = \|X\|_F$ and $f(x) = \|x\|_2$

A natural question arises: what properties do f and F share? Does differentiability of one imply differentiability of the other?, how about convexity?, or even analyticity? Surprisingly the answer to these questions is yes. Indeed, often more can be said about the explicit relationship. Common proofs stem from the well-known trace-inequality [9, Theorem 1.2.1]:

$$\langle \lambda(X), \lambda(Y) \rangle \geq \langle X, Y \rangle$$

with equality if and only if X and Y admit a simultaneous ordered spectral decomposition: there exists an orthogonal matrix U such that

$$UXU^T = \lambda(X) \text{ and } UYU^T = \lambda(Y).$$

The trace inequality yields the elegant relation $F^* = f^* \circ \lambda$ where F^* and f^* are the convex conjugates of F and f, respectively. This equality, in turn, yields the simple representation of the subdifferential of convex functions [60]:

$$\begin{aligned}\partial_P F(X) = \big\{U\mathrm{Diag}(v)U^T :\ U \text{ is orthogonal with } U\mathrm{Diag}(\lambda(X))U^T = X \\ \text{and } v \in \partial_P f(\lambda(X))\big\}.\end{aligned} \tag{1.1}$$

Similar results hold for the Fréchet, limiting, and Clarke subdifferentials. The formula provides a concise way of computing the subdifferentials of non-smooth matrix-valued functions.

Example 1.1.1 (The proximal subdifferential of the nuclear norm)**.** A quick check shows

$$(\partial_P \|x\|_1)_j = \begin{cases} \mathrm{sign}(x_j), & \text{if } |x_j| \neq 0 \\ [-1,1], & \text{otherwise} \end{cases}$$

and thus if $[U_+, U_0, U_-]\mathrm{Diag}(\lambda(X))[U_+, U_0, U_-]^T = X$ is an orthogonal decomposition of X, then

$$\partial_P \|X\|_* = \left\{ [U_+, U_0, U_-] \begin{pmatrix} I & 0 & 0 \\ 0 & W & 0 \\ 0 & 0 & -I \end{pmatrix} [U_+, U_0, U_-] : \|W\|_2 \leq 1 \right\}$$

[112, Example 2].

The content of Chapter 4 is twofold: (1) we show a much simpler proof of (1.1) for non-convex spectral functions and (2) we provide a geometric argument for computing the Hessian by designing special curves on the epigraph of the C^2-smooth spectral function.

1.2 Oracle complexities

In the next few sections, we will describe several algorithms for solving the problem

$$\min_x \ f(x).$$

For this, we want to compare algorithms to each other; in other words we are looking for a metric with which we can say that some algorithm is "better" than the other. For the rest of this thesis, we will use the classical *black-box model*; this model assumes that we can access information about the function f via queries to *oracles.* A typical classification of the oracle is based on the order of the derivative output:

- a *zeroth-order* oracle takes in a point $x \in \mathbb{R}^n$ and outputs the value $f(x)$
- a *first-order* oracle takes in a point $x \in \mathbb{R}^n$ and outputs both the value of the function f at x and the gradient of the function at x
- a *second-order* oracle takes in a point $x \in \mathbb{R}^n$ and outputs $f(x)$, the gradient $\nabla f(x)$, and the Hessian $\nabla^2 f(x)$.

For our purposes, we focus on first-order oracles. Such oracles naturally fit problems arising in machine learning when the dimension of $\mathbb{R}^n$ is large; thus making computation of the Hessians impractical. Our main interests are in the *oracle complexity*, that is how many queries to the oracle are necessary and sufficient to obtain an ε-approximate minima to a convex function. For this, we need both an upper and lower bound; the upper bound comes from constructing a specific algorithm and the lower bound from devising a clever function (or a sequence) and arguing if the number of queries to the oracle is "too small" no algorithm has enough information about the function to know whether an ε-approximate minima has been reached. The black box model was developed in the early days of convex optimization by pioneers Nemirovski and Yudin [78].

Classifying the objective function When comparing complexities of algorithms, we jointly consider the oracle and the class of functions we are minimizing. Two standard assumptions regarding the objective function f are *β-smoothness* and *strong convexity.* We discuss each in detail.

Definition 1.2.1 (β-smoothness)**.** A continuously differentiable $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is *β-smooth* if the gradient $\nabla f(x)$ is Lipschitz with constant β, that is

$$\|\nabla f(x) - \nabla f(y)\| \leq \beta \|x - y\| .$$

Typical algorithms use simple "models" of the objective function as proxies. The β-smoothness condition provides a global quadratic majorization model:

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\beta}{2} \|x - y\|^2, \quad \text{for all } x, y \in \mathbb{R}^n. \tag{1.2}$$

The quadratic model is a simpler object to work with. For instance, by minimizing this quadratic, we directly recover the gradient descent algorithm. For a C^2-smooth function, the β-smoothness reduces to a statement about the Hessian, $\beta I \succeq \nabla^2 f(x) \succeq -\beta I$.

We now discuss another property of the original minimizing function which can significantly speed up convergence, known as strong convexity.

Definition 1.2.2 (α-strongly convex)**.** A function $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is *α-strongly convex* ($\alpha \geq 0$) if it satisfies the subgradient inequality:

$$f(y) \geq f(x) + \langle v, y - x \rangle + \frac{\alpha}{2} \|y - x\|^2, \quad \text{for all } x, y \in \mathbb{R}^n \text{ and } v \in \partial f(x). \tag{1.3}$$

Notice when the $\alpha = 0$, this condition reduces to convexity; so strong convexity strengthens convexity. Parsing Definition 1.2.2, strong convexity says the original function f is lower bounded by a convex quadratic function. We think of the constant α as a measure of the *curvature* of the function; large curvature ($\alpha > 0$) guarantees that f has a minimizer on $\mathbb{R}^n$. It is immediate to verify that a function f is α-strongly convex if and only if $x \mapsto f(x) - \frac{\alpha}{2} \|x\|^2$ is convex (in particular, if f is C^2-smooth and α-convex, then $\nabla^2 f(x) \succeq \alpha I$).

Significant improvements in oracle complexities occur when both the minimizing function has smoothness and strong convexity properties. In particular, the addition of a strong convexity assumption changes the rate of convergence from *sublinear* (here we assume β-smoothness) to *linear rate of convergence* (exponential decay).

1.2.1 Measuring oracle complexities

We have intuitively defined oracle complexities as the number of calls to the oracle to obtain ε-approximate minima; but what does an ε-approximate minima mean? Generally, there are

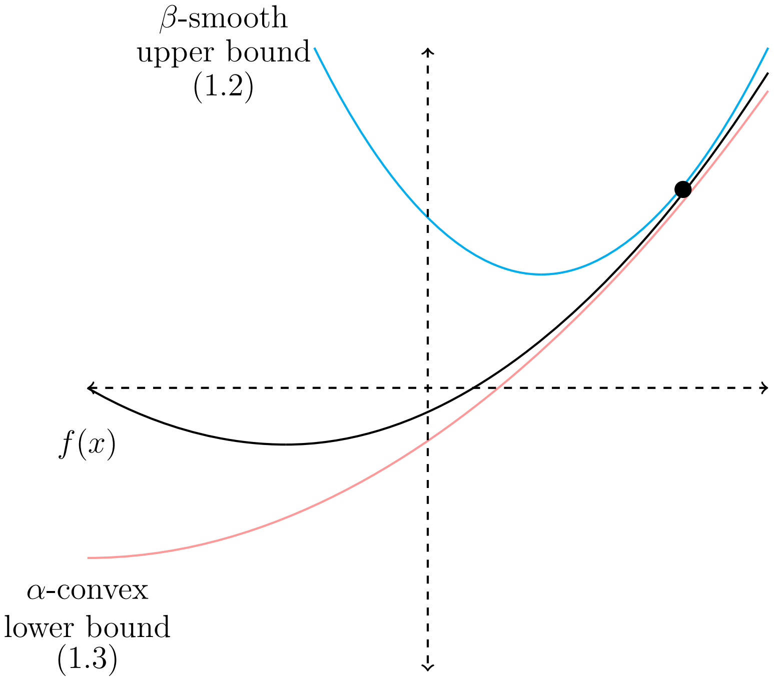


Figure 1.2: Illustration of the upper and lower bounds of an α-convex, β-smooth function

three distinct notions: (1) $f(x) - \inf f < \varepsilon$ (function values), (2) $\|x - x^*\|^2 < \varepsilon$ (iterates), and (3) $\mathrm{dist}(0, \partial f(x)) < \varepsilon$ (subgradients). When the minimizing function is non-convex and we use a first-order oracle, then we expect a first-order algorithm to output a point with a small subgradient. When the function is both strongly convex and β-smooth, complexities with respect to all three measures are all equivalent.

1.3 A brief history of classical first-order methods

Since Chapter 2 and Chapter 3 compare our newly designed algorithms with previous results, we begin by summarizing some of standard methods. Consider the unconstrained problem

$$\min_x \; f(x) \tag{1.4}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a β-smooth function. Let us begin with the *gradient-descent scheme* and its variants. This iterative scheme dates back to Cauchy [22] and is perhaps the simplest strategy for minimizing a differentiable function f on $\mathbb{R}^n$. Starting at some initial point x_0, the scheme iterates

$$x_{t+1} = x_t - \eta \nabla f(x_t). \tag{1.5}$$

The derivation of gradient descent update rule comes from minimizing the quadratic upper bound (1.2). From this majorization viewpoint, it is clear that gradient descent produces iterates with decreasing function values (see Figure 1.3). The value of η, also known as the *step length*, controls how far to move in the negative gradient direction (here $\eta = 1/\beta$). To obtain an ε-stationary point, the gradient descent scheme (1.5) requires at most $O(\beta(f(x_0) - f^*)/\varepsilon^2)$ iterations [81].[1] By setting $\eta = 1/\beta$, a convex and β-smooth objective function requires $O(\beta \left\|x_0 - x^*\right\|^2 / \varepsilon)$ number of iterations to achieve a ε-optimal solution (see [12, Theorem 3.3] or [81, Corollary 2.12]). By assuming a stronger condition, namely α-convexity, the convergence rate becomes linear, $\left\|x_t - x^*\right\|^2 \leq (1 - \frac{\alpha}{\beta})^t \left\|x_0 - x^*\right\|$; see e.g. [81, Theorem 2.1.15].

As we expect, non-smoothness of the minimizing function significantly impacts the convergence guarantees. To see this, we assume the function f is convex and L-Lipschitz ($\|f(x) - f(y)\| \leq L\|x - y\|$ for all x, y) but not β-smooth. In this context, we make a modification to the basic gradient descent (1.5):

$$x_{t+1} = x_t - \eta v_t, \qquad \text{where } v_t \in \partial f(x_t).$$

Known in the literature as the *subgradient descent algorithm*, this method with $\eta = \frac{\|x_0 - x^*\|}{L\sqrt{t}}$ requires $\widetilde{O}(\left\|x_0 - x^*\right\|^2 L^2/\varepsilon^2)$ number of calls to the oracle to achieve an ε-optimal point (small function values) [12, Theorem 3.2].

As illustrated above, the effects of convexity and smoothness on the convergence rates is astonishing. On the one hand, the gradient descent method is the workhorse of first-order

[1] The big O notation hides only problem independent constants and $f^* = \lim_{t\to\infty} f(x_t)$.

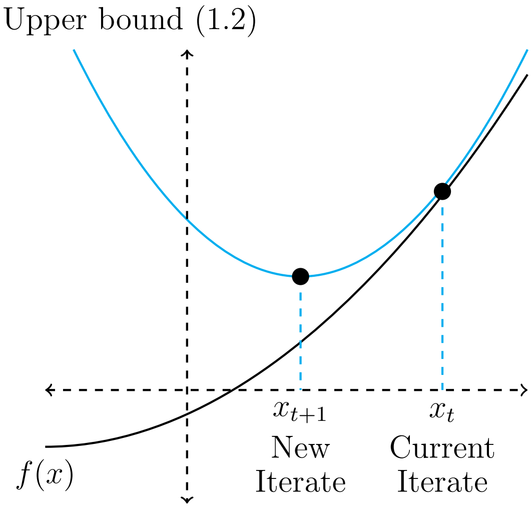


Figure 1.3: Majorization view of gradient descent for β-smooth functions

optimization and it produces guarantees *independent* of the dimension of the space under various assumptions on the minimizing function. On the other hand, it emphasizes the importance of convexity and smoothness with number of oracle calls ranging from sublinear rates $O(1/\varepsilon^2)$ and $O(1/\varepsilon)$ to linear rates $O(\frac{\beta}{\alpha}\log(1/\varepsilon))$.

1.3.1 An interlude into first-order oracle lower bounds

At this point, one may be wondering if gradient descent gives the best convergence guarantees for a first-order oracle for smooth minimization. As mentioned in Section 1.2, the oracle complexity is governed by both an upper and lower bound. So far, we have only discussed possible upper bounds arising from gradient descent. Let us discuss lower-bounds. These results originally appeared in Nemirovski and Yudin [78] and later a simplified version appeared in Nesterov [81].

One way to think about a black box algorithm is that it has available a “history” of the points and gradients that have been generated so far and it outputs a new point based only on this information, which will be fed into the first-order oracle. Therefore, let us make the following simplifying assumption on the iterate sequence:

$$x_{t+1} \in x_0 + \mathrm{Span}\{v_0, v_1, \ldots, v_t\}, \tag{1.6}$$

with $v_i \in \partial f(x_i)$ for each index i. Under this assumption, we obtain three lower bounds:

Theorem 1.3.1 (Lower bounds on the first-order oracles)**.** *Assume the black box procedure satisfies* (1.6) *. Then we have the following three lower bounds:*

(1) (Convex and Non-smooth) Let $t \leq n$ and R any positive real constant. There exists a convex and L-Lipschitz function f such that

$$\min_{1 \leq s \leq t} f(x_s) - \min_{x \in B(0,R)^2} f(x) \geq \frac{RL}{2(1+\sqrt{t})}.$$

(2) (Convex and smooth) Let $t \leq (n-1)/2, \beta > 0$. There exists a β-smooth convex function f such that

$$\min_{1 \leq s \leq t} f(x_s) - f(x^*) \geq \frac{3\beta}{32} \frac{\|x_1 - x^*\|^2}{(t+1)^2}.$$

(3) (Strongly Convex and Smooth) There exists a β-smooth and α-strongly convex function f such that for any $t \geq 1$ one has

$$f(x_t) - f(x^*) \geq \frac{\alpha}{2} \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^{2(t-1)} \|x_1 - x^*\|^2, \quad \textit{where } \kappa = \beta/\alpha.$$

The proof of these statements can be found in [12, Theorem 3.13, Theorem 3.14, Theorem 3.15] and [81, Theorem 2.1.7].

Observe the gap in the upper and lower-complexity bounds for smooth convex minimization: gradient descent achieves the efficiency estimate $O(1/\varepsilon)$ (and $O(\frac{\beta}{\alpha} \log(1/\varepsilon))$ in the strongly convex case) while the lower bounds suggest complexities of $O(1/\sqrt{\varepsilon})$ (and $O(\sqrt{\frac{\beta}{\alpha}} \log(1/\varepsilon)$ in the strongly convex case). Historically, the first method matching *optimal oracle complexity* was inspired by the conjugate gradient algorithm for solving systems of equations [78]. Nesterov, in the 1980s, introduced the first true first-order optimal algorithm

[81, 82]. The latter algorithm, known as *Nesterov' accelerated gradient method* is one of today's most influential algorithms for smooth optimization. Following this, there have been a slew of optimal methods. However, one open result remains: as far as we are aware, no lower bound on the complexity exists for minimizing smooth non-convex functions. The gradient descent rate of $O(1/\varepsilon^2)$ is the "best" currently available rate.

		# of Iterations			
Method	convergence guarantee	non-convex, β-smooth	convex, β-smooth	α-convex, β-smooth	cost/iter
(Prox) Gradient	ε-stationary	$O\big(\frac{\beta}{\varepsilon^2}\big)$	$O\big(\frac{\beta}{\varepsilon}\big)$	$O\big(\frac{\beta}{\alpha}\log(\frac{1}{\varepsilon})\big)$	$1\ \nabla c(x)$
	ε-optimal	x	$O\big(\frac{\beta}{\varepsilon}\big)$	$O\big(\frac{\beta}{\alpha}\log(\frac{1}{\varepsilon})\big)$	$1\ \nabla c(x)$
Accelerated	ε-stationary	x	$O\big(\frac{\beta^{2/3}}{\varepsilon^{2/3}}\big)$	$O\big(\sqrt{\frac{\beta}{\alpha}}\log(\frac{1}{\varepsilon})\big)$	$1\ \nabla c(x)$
	ε-optimal	x	$O\big(\sqrt{\frac{\beta}{\varepsilon}}\big)$	$O\big(\sqrt{\frac{\beta}{\alpha}}\log(\frac{1}{\varepsilon})\big)$	$1\ \nabla c(x)$
Stochastic[3]	ε-stationary	x	x	$O\big((m+\frac{\beta}{\alpha})\log(\frac{1}{\varepsilon})\big)$	$1\ \nabla c_i(x)$
	ε-optimal	x	$O\big(\frac{m+\beta}{\varepsilon}\big)$[4]	$O\big((m+\frac{\beta}{\alpha})\log(\frac{1}{\varepsilon})\big)$[5]	$1\ \nabla c_i(x)$

Table 1.1: Convergence rates for first-order methods

1.3.2 Additive Composite Functions

A striking difference in complexities exists due to non-smoothness in the blackbox model. In the absence of smoothness, we use the *structure* of the objective function to provide better

[3] The function $f(x) = \frac{1}{m}\sum_{i=1}^{m} c_i(x) + g(x)$

[4] The convergence rate is stated for SAGA. [34]

[5] The rate is stated for SVRG. [55]

rates of convergence for these non-smooth problems. A well-studied structured optimization problem known as *additive composite minimization*, or simply *composite functions*, in the literature is formulated as

$$\min_{x} \; f(x) = c(x) + g(x). \tag{1.7}$$

The convergence analysis has been well-studied in the literature under various contexts and assumptions on the functions c and g (see [4, 81, 82, 109]). Usual formulations assume the functions $c(x)$ and $g(x)$ are convex and $c(x)$ is β-smooth. The methods in Chapter 2 and Chapter 3 use many of the ideas developed for this problem class; as such we provide a brief survey of suboptimal and optimal methods for solving (1.7).

Algorithms for additive composite minimization A common approach for solving (1.7) are the *proximal gradient methods* (see [96]). To motivate this algorithm, consider a constrained optimization problem: minimizing a β-smooth, convex function $c(x)$ over a closed convex set C (take $g(x)$ in (1.7) to be the indicator of the set C). Imitating the gradient descent algorithm (1.5), we take the gradient-step as before, however the new point may be infeasible so we rectify this by projecting back onto to the set C:

$$x_{k+1} = \mathrm{Proj}_C(x_k - \tfrac{1}{\beta}\nabla c(x_k)) = \operatorname*{argmin}_{x \in C} \; \frac{1}{2}\left\| x - (x_k - \tfrac{1}{\beta}\nabla c(x_k)) \right\|^2 .$$

When the function g is a general convex function, we replace the projection operation with the *proximal mapping*:

$$\mathrm{prox}_{tg}(w) = \operatorname*{argmin}_{x} \; \left\{ g(x) + \frac{1}{2t}\left\| x - w \right\|^2 \right\}. \tag{1.8}$$

When this subproblem is inexpensive, the *prox-gradient algorithm* is simply

$$x_{k+1} = \mathrm{prox}_{tg}(x_k - \tfrac{1}{\beta}\nabla c(x_k)). \tag{1.9}$$

The proximal gradient method enjoys oracle complexity similar to gradient descent: $O(\varepsilon^{-1})$ for an ε-optimal solution when f is convex and $O(\varepsilon^{-2})$ for an ε-stationary solution when c is β-smooth but not convex.

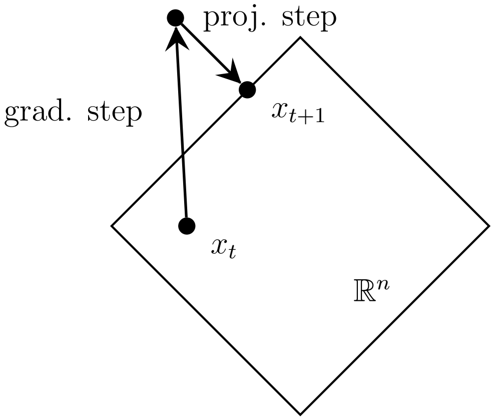


Figure 1.4: Illustration of projected gradient descent

The optimal method introduced by Nesterov adaptes to the additive composite minimization problem when the function $c(x)$ is convex (See Figure 1.3.2 and Figure 1.3.2). The variant, discovered by Beck and Tebulle [4], achieves the same optimal rates as the smooth cases ($O(\sqrt{\beta/\varepsilon})$ and $O(\sqrt{\beta/\alpha}\log(1/\varepsilon))$ in the convex and α-convex cases, respectively).

Incremental Methods In machine learning applications, structured optimization problems often take the form:

$$\min_x f(x) := \frac{1}{m}\sum_{i=1}^{m} c_i(x) + g(x). \tag{1.10}$$

where the functions c_i are convex and β-smooth, the function $g(x)$ is known as the regularizer, and m is large. Computing a full gradient of (1.10) when m is large is expensive; however by introducing randomness, convergence rates are retained on *average*. These methods are known as *stochastic*. The simplest stochastic method is *stochastic gradient descent* (SGD). We replace at each step the full gradient in the prox-gradient method (1.9) with a uniformly chosen gradient $\nabla c_{i_k}(x)$ where $i_k \in [n]$. Using a decreasing step-size η, on average an ε-approximate optimal solution requires $O(1/\varepsilon^2)$ iterations when the minimizing function is convex and $O(1/(\alpha\varepsilon))$ iterations when the function is α-strongly convex (See Table 1.1 for

```
**Input:** Choose $x_0 \in \operatorname{dom} f$ and $a_0 \in (0,1)$.
**Initialization:** $y_0 = x_0$
**Step k** $(k = 1, 2, \ldots)$
    1. *(Proximal Step)* Find $x_k$ such that
           $$x_k = \operatorname{prox}_{\beta^{-1}g}\left(y_k - \frac{1}{\beta}\nabla c(y_k)\right).$$
    2. Choose $a_{k+1} \in (0,1)$ satisfying
           $$a_{k+1}^2 = a_k^2(1 - a_{k+1}).$$
       and set $b_k = \frac{a_k(1-a_k)}{a_k^2 + a_{k+1}}$.
    3. *(Momentum Step)* Update
           $$y_{k+1} = x_{k+1} + b_k(x_{k+1} - x_k).$$
```

```
**Input:** Choose $x_0 \in \operatorname{dom} f$ and $a_0 \in \left[\sqrt{\frac{\alpha}{\beta}}, 1\right)$
**Initialization:** $y_0 = x_0$ and $q = \frac{\alpha}{\beta}$
**Step k** $(k = 1, 2, \ldots)$
    1. *(Proximal Step)* Find $x_k$ such that
           $$x_{k+1} = \operatorname{prox}_{\beta^{-1}g}\left(y_k - \frac{1}{\beta}\nabla c(y_k)\right).$$
    2. Compute $a_{k+1} \in (0,1)$ from the equation
           $$a_{k+1}^2 = (1 - a_{k+1})a_k^2 + q \cdot a_{k+1}$$
       and set $b_k = \frac{a_k(1-a_k)}{a_k^2 + a_{k+1}}$.
    3. *(Momentum Step)* Update
           $$y_{k+1} = x_{k+1} + b_k(x_{k+1} - x_k).$$
```

Figure 1.5: Pseudocode for Nesterov's accelerated gradient method in the non-strongly convex case ($\alpha = 0$) (left) and strongly convex case (right)

comparison with full gradient computations). So-called *incremental methods* achieve faster rates, and include SAG (Stochastic Averaged Gradient), SAGA, SDCA (Stochastic Dual Coordinate Ascent), and SVRG (Stochastic Variance Reduced Gradient descent). Incremental methods require $O((m + \beta/\alpha)\log(1/\varepsilon))$ gradient computations when f is α-strongly convex to achieve ε accuracy [34, 55, 99, 103]. We describe SVRG in Algorithm 1.

1.4 Inertial acceleration beyond convexity

A major part of the thesis concerns the idea of "acceleration". The term, acceleration, applies to first-order methods whose efficiency guarantees match the optimal lower complexity estimates within the problem class. A goal of this thesis is to extend this notion of "acceleration" to bridge between convex and non-convex settings. We desire a single first-order

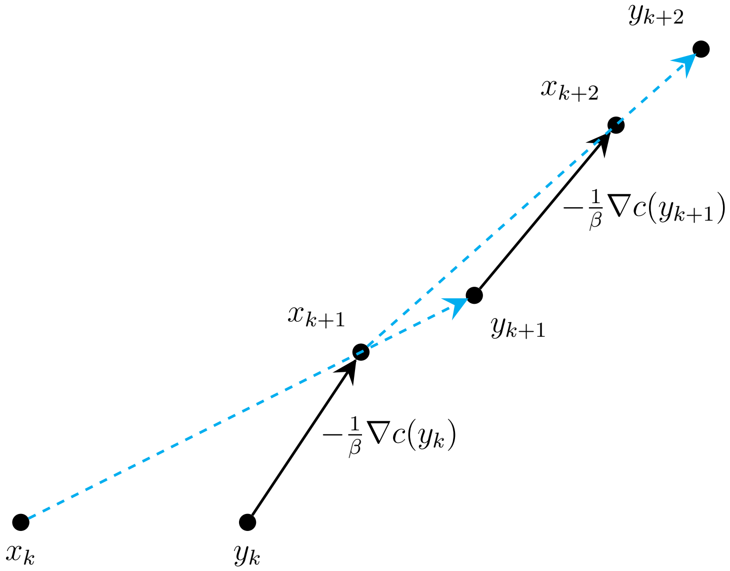

y_{k+2}
x_{k+2}
$-\frac{1}{\beta}\nabla c(y_{k+1})$
x_{k+1}
y_{k+1}
$-\frac{1}{\beta}\nabla c(y_k)$
x_k
y_k

Figure 1.6: Illustration of Nesterov's accelerated gradient descent for $\min_x c(x)$.

method that achieves best known rates for convex and non-convex problems simultaneously. In particular, we focus on when the user is unaware of the convexity of the objective. The idea stems from the works of Ghadimi and Lan [48].

In Chapter 2, we explore a variety of algorithms for optimizing a class of functions called *convex composite functions*, and having the form $h(c(x)) + g(x)$, where h is Lipschitz and convex, c is smooth, and g is simply convex. When h is the identity, this problem reduces to the additive composite minimization problem (see 1.3.2). The composition structure allows us to construct algorithms for these minimization problems. Using the prox-gradient method as our model, the *prox-linear method* iterates:

$$x_{k+1} = \operatorname*{argmin}_x \left\{ h\big(c(x_k) + \nabla c(x_k)(x - x_k)\big) + g(x) + \frac{1}{2t}\left\|x - x_k\right\|^2 \right\}. \tag{1.11}$$

By combining this result with Nesterov's accelerated method, we construct a single "accelerated" method that is adaptive to underlying convexity of the problem. In Chapter 3, we return to the finite-sum problem (1.10), but now assume the functions c_i are non-convex.

```
Algorithm 1: SVRG
  input: Fix point $y^{(1)} \in \mathbb{R}^n$.
  repeat for $s = 1, 2, \ldots$
    1. initialization: Set $x_1^{(s)} = y^{(s)}$
    2. repeat for $t = 1, \ldots, k$
      (a) Choose $i_t^{(s)}$ uniformly at random from $[m]$
      (b) Compute
            $x_{t+1}^{(s)} = x_t^{(s)} - \eta \left( \nabla f_{i_t^{(s)}}(x_t^{(s)}) - \nabla f_{i_t^{(s)}}(y^{(s)}) + \nabla f(y^{(s)}) \right)$
    3. SetT
            $y^{(s+1)} = \frac{1}{k} \sum_{t=1}^{k} x_t^{(s)}.$
```

Under this new setting, we also construct a generic “accelerated” method.

Both Chapters 2 and 3 rely on minimizing a wide class of non-convex functions, that are convex up to a perturbation. This function class is called *weakly-convex.*

Definition 1.4.1 (Weak convexity)**.** A function $f \colon \mathbb{R}^p \to \mathbb{R} \cup \{\infty\}$ is $\rho-$*weakly convex* if for any points $x, y \in \mathbb{R}^p$ and $\lambda \in [0, 1]$, the approximate secant inequality holds:

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) + \rho\lambda(1-\lambda) \left\| x - y \right\|^2 .$$

First, notice that a ρ-weakly convex function with ρ equal to 0 is simply convex. Alternatively, easy algebraic manipulations show that a function is ρ-weakly convex if and only if $x \mapsto f(x) + \frac{\rho}{2} \left\| x \right\|^2$ is convex. In particular, a C^1-smooth function f is ρ-weakly convex if the gradient ∇f is ρ-Lipschitz, while a C^2-smooth function f is ρ-weakly convex if and only if $\nabla^2 f(x) \succeq -\rho I$ for all x. In this sense the constant ρ is a measure of convexity; a function behaves more like a convex function when ρ is small.

Example 1.4.1 (Sum of ρ-weakly convex functions)**.** Consider an additive composite minimization problem

$$\min_x \; f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x) + g(x),$$

where $f_i \colon \mathbb{R}^p \to \mathbb{R} \cup \{\infty\}$ are ρ_i-weakly convex and $g \colon \mathbb{R}^p \to \mathbb{R} \cup \{\infty\}$ is ρ-weakly convex. In typical applications, the first summand measures fidelity of the predicted response to observed data (e.g. penalty on misfit, log-likelihood) and g is a structure inducing regularizer on the covariates x. From the very definition of weak convexity, it is clear that f is $(\rho + \sum_{i=1}^n \rho_i/n)$-weakly convex. Departure from true convexity is common, since the loss functions f_i can easily be non-convex but smooth with Lipschitz continuous gradients. Similarly, it is common to use non-convex (in fact, concave) regularizers g to induce sparsity for example.

Example 1.4.2 (Fully composite problems)**.** Another rich class of problems that are weakly convex consists of fully composite models

$$\min_x \; f(x) := \frac{1}{n}\sum_{i=1}^{n} h_i(c_i(x)) + g(x)$$

where $h_i \colon \mathbb{R}^q \to \mathbb{R}$ are L_i-Lipschitz and convex, $c_i \colon \mathbb{R}^p \to \mathbb{R}^{q_i}$ are C^1-smooth with ρ_i-Lipschitz Jacobian ∇c_i, and $g \colon \mathbb{R}^p \to \mathbb{R} \cup \{\infty\}$ is ρ-weakly convex. Then f is $(\rho + \sum_{i=1}^n L_i \rho_i/n)$-weakly convex (Lemma 4.4 in [39]).

Weakly convex functions have appeared in a wide variety of contexts, and under different names. Some notable examples are globally lower-C^2 [97], prox-regular [91], proximally smooth functions [25], and those functions whose epigraph has positive reach [45]. In Chapter 2, we explore the complexity of minimizing convex composite problems, while in Chapter 3, we describe a generic schema for “accelerating” existing algorithms for weakly-convex finite sum problems.

At the beginning of each chapter, we outline a precise description of the work done.

Chapter 2

EFFICIENCY OF MINIMIZING COMPOSITIONS OF CONVEX FUNCTIONS AND SMOOTH MAPS

Joint work with D. Drusvyatskiy [39]

Abstract. We consider global efficiency of algorithms for minimizing a sum of a convex function and a composition of a Lipschitz convex function with a smooth map. The basic algorithm we rely on is the prox-linear method, which in each iteration solves a regularized subproblem formed by linearizing the smooth map. When the subproblems are solved exactly, the method has efficiency $\mathcal{O}(\varepsilon^{-2})$, akin to gradient descent for smooth minimization. We show that when the subproblems can only be solved by first-order methods, a simple combination of smoothing, the prox-linear method, and a fast-gradient scheme yields an algorithm with complexity $\widetilde{\mathcal{O}}(\varepsilon^{-3})$. The technique readily extends to minimizing an average of m composite functions, with complexity $\widetilde{\mathcal{O}}(m/\varepsilon^2 + \sqrt{m}/\varepsilon^3)$ in expectation. We round off the paper with an inertial prox-linear method that automatically accelerates in presence of convexity.

2.1 Introduction

In this work, we consider the class of *composite optimization problems*

$$\min_x \; F(x) := g(x) + h(c(x)), \tag{2.1}$$

where $g\colon \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ and $h\colon \mathbb{R}^m \to \mathbb{R}$ are closed convex functions and $c\colon \mathbb{R}^d \to \mathbb{R}^m$ is a smooth map. Classical examples include regularized nonlinear least squares [87, Section 10.3] and exact penalty formulations of nonlinear programs [87, Section 17.2], while notable contemporary instances include robust phase retrieval [41, 42] and matrix factorization problems such as NMF [50]. The setting where c maps to the real line and h is the identity function, namely

$$\min_{x} \; c(x) + g(x), \tag{2.2}$$

is now commonplace in large-scale optimization. In this work, we use the term *additive composite minimization* for (2.2) to distinguish it from the more general composite class (2.1).

The most basic first-order method for additive composite minimization is the prox-gradient algorithm, investigated by Beck-Teboulle [4] and Nesterov [86, Section 3]. Similarly, much of the current paper will center around the *prox-linear method* – a direct extension of the prox-gradient algorithm to the entire problem class (2.1). In each iteration, the prox-linear method linearizes the smooth map $c(\cdot)$ and solves the *proximal subproblem*:

$$x_{k+1} = \operatorname*{argmin}_{x} \left\{ g(x) + h\Big(c(x_k) + \nabla c(x_k)(x - x_k)\Big) + \tfrac{1}{2t}\|x - x_k\|^2 \right\}, \tag{2.3}$$

for an appropriately chosen parameter $t > 0$. The underlying assumption here is that the strongly convex proximal subproblems (2.3) can be solved efficiently. This is indeed reasonable in some circumstances. For example, one may have available specialized methods for the proximal subproblems, or interior-point points methods may be available for moderate dimensions d and m, or it may be that case that computing an accurate estimate of $\nabla c(x)$ may already be the bottleneck (see e.g. Example 2.3.5). The prox-linear method was recently investigated in [20, 38, 66, 83], though the ideas behind the algorithm and of its trust-region variants are much older [13, 20, 46, 92, 93, 115, 117]. The scheme (2.3) reduces to the popular prox-gradient algorithm for additive composite minimization, while for nonlinear least squares, the algorithm is closely related to the Gauss-Newton algorithm [87, Section 10].

Our work focuses on global efficiency estimates of numerical methods. Therefore, in line with standard assumptions in the literature, we assume that h is L-Lipschitz and the Jacobian map ∇c is β-Lipschitz. As in the analysis of the prox-gradient method in Nesterov [81, 82], it is convenient to measure the progress of the prox-linear method in terms of the scaled step-sizes, called the *prox-gradients*:

$$\mathcal{G}_t(x_k) := t^{-1}(x_k - x_{k+1}).$$

A short argument shows that with the optimal choice $t = (L\beta)^{-1}$, the prox-linear algorithm will find a point x satisfying $\|\mathcal{G}_{\frac{1}{L\beta}}(x)\| \le \varepsilon$ after at most $\mathcal{O}(\frac{L\beta}{\varepsilon^2}(F(x_0) - \inf F))$ iterations; see e.g. [20, 38]. We mention in passing that iterate convergence under the KŁ-inequality was recently shown in [8, 90], while local linear/quadratic rates under appropriate regularity conditions were proved in [16, 38, 83]. The contributions of our work are as follows.

1. **(Prox-gradient and the Moreau envelope)** The size of the prox-gradient $\|\mathcal{G}_t(x_k)\|$ plays a basic role in this work. In particular, all convergence rates are stated in terms of this quantity. Consequently, it is important to understand precisely what this quantity entails about the quality of the point x_k (or x_{k+1}). For additive composite problems (2.2), the situation is clear. Indeed, the proximal gradient method generates iterates satisfying $F'(x_{k+1}; u) \ge -2\|\mathcal{G}_{\frac{1}{\beta}}(x_k)\|$ for all unit vectors u, where $F'(x; u)$ is the directional derivative of F at x in direction u [86, Corollary 1]. Therefore, a small prox-gradient $\|\mathcal{G}_{\frac{1}{\beta}}(x_k)\|$ guarantees that x_{k+1} is nearly stationary for the problem, since the derivative of F at x_{k+1} in any unit direction is nearly nonnegative. For the general composite class (2.1), such a conclusion is decisively false. Case in point, the prox-linear method can generate an iterate sequence along which F is differentiable with gradient norms $\|\nabla F(x_k)\|$ uniformly bounded away from zero, in spite of the norms $\|\mathcal{G}_{\frac{1}{L\beta}}(x_k)\|$ tending to zero.[1] Therefore, we must justify our focus on the norm $\|\mathcal{G}_{\frac{1}{L\beta}}(x_k)\|$ by other means. Our first contribution is Theorem 2.4.5: we prove that $\|\mathcal{G}_{\frac{1}{L\beta}}(x)\|$ is proportional to the norm of the true gradient of the Moreau envelope of F — a well studied smooth approximation of F having identical stationary points. An immediate consequence is that even though x might not be nearly stationary for F, a small prox-gradient $\|\mathcal{G}_{\frac{1}{L\beta}}(x)\|$ guarantees that x is near some point $\hat{x}$ (the proximal

[1] See the beginning of Section 2.4 for a simple example of this type of behavior.

point), which is nearly stationary for F. In this sense, a small prox-gradient $\|\mathcal{G}_{\frac{1}{L\beta}}(x)\|$ is informative about the quality of x. We note that an earlier version of this conclusion based on a more indirect argument, appeared in [38, Theorem 5.3], and was used to derive linear/quadratic rates of convergence for the prox-linear method under suitable regularity conditions.

2. **(Inexactness and complexity of first-order methods)** For the general composite class (2.1), coping with inexactness in the proximal subproblem solves (2.3) is unavoidable. We perform an inexact analysis of the prox-linear method based on two natural models of inexactness: (*i*) near-optimality in function value and (*ii*) near-stationarity in the dual. Based on the inexact analysis, it is routine to derive overall efficiency estimates for the prox-linear method, where the proximal subproblems are themselves solved by first-order algorithms. We do not record such estimates in this paper; instead, we present algorithms based on a smoothing technique, for which we can prove better efficiency estimates.

3. **(Improved complexity of first-order methods through smoothing)** Smoothing is a common technique in nonsmooth optimization. The seminal paper of Nesterov [82], in particular, derives convergence guarantees for algorithms based on infimal convolution smoothing in structured convex optimization. In contrast, we are not aware of any worst-case global efficiency estimates based on smoothing for nonconvex problems. In the context of the composite class (2.1), smoothing is indeed appealing. In the simplest case, one replaces the function h by a smooth approximation and solves the resulting smooth problem instead.

   We advocate running an inexact prox-linear method on the smooth approximation, with the proximal subproblems approximately solved by fast-gradient methods. To state the resulting complexity bounds, let us suppose that there is a finite upper bound on the operator norms $\|\nabla c(x)\|_{\text{op}}$ over all x in the domain of g, and denote it by $\|\nabla c\|$.

We prove that the outlined scheme requires at most

$$\widetilde{\mathcal{O}}\left(\frac{L^2\beta\|\nabla c\|}{\varepsilon^3}(F(x_0)-\inf F)\right) \tag{2.4}$$

matrix vector products $\nabla c(x)v$, $\nabla c(x)^T w$ and proximal operations of g and h to find a point x satisfying $\|\mathcal{G}_{\frac{1}{L\beta}}(x)\| \leq \varepsilon$. To the best of our knowledge, this is the best known complexity bound for the problem class (2.1) among first-order methods.

4. (**Complexity of finite-sum problems)** Common large-scale problems in machine learning and high dimensional statistics lead to minimizing a large finite sum of functions. Consequently, we consider the finite-sum extension of the composite problem class,

$$\min_x \ F(x) := \frac{1}{m}\sum_{i=1}^m h_i(c_i(x)) + g(x),$$

where now each h_i is L-Lipschitz and each c_i is C^1-smooth with β-Lipschitz gradient. Clearly, the finite-sum problem is itself an instance of (2.1) under the identification $h(z_i,\ldots,z_m) := \frac{1}{m}\sum_{i=1}^m h_i(z_i)$ and $c(x) := (c_1(x),\ldots,c_m(x))$. In this structured context, however, the complexity of an algorithm is best measured in terms of the number of individual gradient evaluations $\nabla c_i(x)$ the algorithm needs to find a point x satisfying $\|\mathcal{G}_{\frac{1}{L\beta}}(x)\| \leq \varepsilon$. A routine computation shows that the efficiency estimate (2.4) of the basic inexact prox-linear method described above leads to the complexity

$$\mathcal{O}\left(\frac{m\cdot L^2\beta\|\nabla c\|}{\varepsilon^3}\left(F(x_0)-\inf F\right)\right) \tag{2.5}$$

in terms of individual gradient evaluations, where now $\|\nabla c\|$ is defined to be an upper bound on $\|\nabla c_i(x)\|$ over all $i = 1,\ldots,m$ and $x \in \operatorname{dom} g$. We show that a better complexity in expectation is possible by incorporating (accelerated)-incremental methods [1, 47, 56, 69, 104] for the proximal subproblems. The resulting algorithm will generate a point x satisfying

$$\mathbb{E}[\|\mathcal{G}_{\frac{1}{L\beta}}(x)\|] \leq \varepsilon,$$

after at most

$$\mathcal{O}\left(\left(\frac{mL\beta}{\varepsilon^2} + \frac{\sqrt{m}\cdot L^2\beta\|\nabla c\|}{\varepsilon^3}\right)\cdot (F(x_0) - \inf F)\right)$$

individual gradient evaluations ∇c_i. Notice that the coefficient of $1/\varepsilon^3$ scales at worst as $\sqrt{m}$ — a significant improvement over (2.5). A different and complementary approach, generalizing stochastic subgradient methods, has been recently pursued by Duchi-Ruan [42].

5. **(Acceleration)** The final contribution of the paper concerns acceleration of the (exact) prox-linear method. For additive composite problems, with c in addition convex, the prox-gradient method is suboptimal from the viewpoint of computational complexity [78, 81]. Accelerated gradient methods, beginning with Nesterov [79] and extended by Beck-Teboulle [4] achieve a superior rate in terms of function values. Later, Nesterov in [85, Page 11, item 2] showed that essentially the same accelerated schemes also achieve a superior rate of $\mathcal{O}((\frac{\beta}{\varepsilon})^{2/3})$ in terms of stationarity, and even a faster rate is possible by first regularizing the problem [85, Page 11, item 3].[2] Consequently, desirable would be an algorithm that *automatically* accelerates in presence of convexity, while performing no worse than the prox-gradient method on nonconvex instances. In the recent manuscript [48], Ghadimi and Lan described such a scheme for additive composite problems. Similar acceleration techniques have also been used for exact penalty formulations of nonlinear programs (2.1) with numerical success, but without formal justification; the paper [15] is a good example.

   We extend the accelerated algorithms of Ghadimi-Lan [48] for additive composite problems to the entire problem class (2.1), with inexact subproblem solves. Assuming the diameter $M := \operatorname{diam}(\operatorname{dom} g)$ is finite, the scheme comes equipped with the guarantee

$$\min_{j=1,\ldots,k}\left\|\mathcal{G}_{\frac{1}{2L\beta}}(x_j)\right\|^2 \le (L\beta M)^2\cdot\mathcal{O}\left(\frac{1}{k^3}+\frac{c_2}{k^2}+\frac{c_1}{k}\right),$$

[2] The short paper [86] only considered smooth unconstrained minimization; however, a minor modification of the proof technique extends to the convex additive composite setting.

where the constants $0 \leq c_1 \leq c_2 \leq 1$ quantify "convexity-like behavior" of the composition. The analysis of the proposed inexact accelerated method based on functional errors shares many features with [100] for convex additive composite problems (2.2).

The outline of the manuscript is as follows. Section 2.2 records basic notation that we use throughout the paper. In Section 2.3, we introduce the composite problem class, first-order stationarity, and the basic prox-linear method. Section 2.4 discusses weak-convexity of the composite function and the relationship of the prox-gradient with the gradient of the Moreau envelope. Section 2.5 analyzes inexact prox-linear methods based on two models of inexactness: near-minimality and dual near-stationarity. In Section 2.6, we derive efficiency estimates of first-order methods for the composite problem class, based on a smoothing strategy. Section 2.7 extends the aforementioned results to problems where one seeks to minimize an average of the composite functions. The final Section 2.8 discusses an inertial prox-linear algorithm that is adaptive to convexity.

2.2 Notation

The notation we follow is standard. Throughout, we consider a Euclidean space, denoted by $\mathbb{R}^d$, with an inner product $\langle \cdot, \cdot \rangle$ and the induced norm $\|\cdot\|$. Given a linear map $A \colon \mathbb{R}^d \to \mathbb{R}^l$, the adjoint $A^* \colon \mathbb{R}^l \to \mathbb{R}^d$ is the unique linear map satisfying

$$\langle Ax, y \rangle = \langle x, A^* y \rangle \qquad \text{for all } x \in \mathbb{R}^d, y \in \mathbb{R}^l.$$

The operator norm of A, defined as $\|A\|_{\text{op}} := \max\limits_{\|u\| \leq 1} \|Au\|$, coincides with the maximal singular value of A and satisfies $\|A\|_{\text{op}} = \|A^*\|_{\text{op}}$. For any map $F \colon \mathbb{R}^d \to \mathbb{R}^m$, we set

$$\text{lip}\,(F) := \sup_{x \neq y} \frac{\|F(y) - F(x)\|}{\|y - x\|}.$$

In particular, we say that F is L-Lipschitz continuous, for some real $L \geq 0$, if the inequality $\text{lip}\,(F) \leq L$ holds. Given a set Q in $\mathbb{R}^d$, the *distance* and *projection* of a point x onto Q are given by

$$\text{dist}(x; Q) := \inf_{y \in Q} \|y - x\|, \qquad \text{proj}(x; Q) := \operatorname*{argmin}_{y \in Q} \|y - x\|,$$

respectively. The extended-real-line is the set $\overline{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$. The *domain* and the *epigraph* of any function $f\colon \mathbb{R}^d \to \overline{\mathbb{R}}$ are the sets

$$\operatorname{dom} f := \{x \in \mathbb{R}^d : f(x) < +\infty\}, \qquad \operatorname{epi} f := \{(x,r) \in \mathbb{R}^d \times \mathbb{R} : f(x) \le r\},$$

respectively. We say that f is *closed* if its epigraph, $\operatorname{epi} f$, is a closed set. Throughout, we will assume that all functions that we encounter are *proper*, meaning they have nonempty domains and never take on the value $-\infty$. The indicator function of a set $Q \subseteq \mathbb{R}^d$, denoted by δ_Q, is defined to be zero on Q and $+\infty$ off it.

Given a convex function $f\colon \mathbb{R}^d \to \overline{\mathbb{R}}$, a vector v is called a *subgradient* of f at a point $x \in \operatorname{dom} f$ if the inequality

$$f(y) \ge f(x) + \langle v, y - x\rangle \qquad \text{holds for all } y \in \mathbb{R}^d. \tag{2.6}$$

The set of all subgradients of f at x is denoted by $\partial f(x)$, and is called the *subdifferential* of f at x. For any point $x \notin \operatorname{dom} f$, we set $\partial f(x)$ to be the empty set. With any convex function f, we associate the *Fenchel conjugate* $f^\star\colon \mathbb{R}^d \to \overline{\mathbb{R}}$, defined by

$$f^\star(y) := \sup_x \{\langle y, x\rangle - f(x)\}.$$

If f is closed and convex, then equality $f = f^{\star\star}$ holds and we have the equivalence

$$y \in \partial f(x) \qquad \Longleftrightarrow \qquad x \in \partial f^\star(y). \tag{2.7}$$

For any function f and real $\nu > 0$, the *Moreau envelope* and the *proximal mapping* are defined by

$$\begin{aligned} f_\nu(x) &:= \inf_z \left\{ f(z) + \frac{1}{2\nu}\|z - x\|^2 \right\}, \\ \operatorname{prox}_{\nu f}(x) &:= \operatorname*{argmin}_z \left\{ f(z) + \frac{1}{2\nu}\|z - x\|^2 \right\}. \end{aligned}$$

respectively. In particular, the Moreau envelope of an indicator function δ_Q is simply the map $x \mapsto \frac{1}{2\nu}\operatorname{dist}^2(x; Q)$ and the proximal mapping of δ_Q is the projection $x \mapsto \operatorname{proj}(x; Q)$. The following lemma lists well-known regularization properties of the Moreau envelope.

Lemma 2.2.1 (Regularization properties of the envelope)**.** *Let $f\colon \mathbb{R}^d \to \mathbb{R}$ be a closed, convex function. Then f_ν is convex and C^1-smooth with*

$$\nabla f_\nu(x) = \nu^{-1}(x - \mathit{prox}_{\nu f}(x)) \qquad \textit{and} \qquad \mathrm{lip}\,(\nabla f_\nu) \leq \tfrac{1}{\nu}.$$

If in addition f is L-Lipschitz, then the envelope $f_\nu(\cdot)$ is L-Lipschitz and satisfies

$$0 \leq f(x) - f_\nu(x) \leq \frac{L^2\nu}{2} \qquad \textit{for all } x \in \mathbb{R}^d. \tag{2.8}$$

Proof. The expression $\nabla f_\nu(x) = \nu^{-1}(x - \mathrm{prox}_{\nu f}(x)) = \nu^{-1} \cdot \mathrm{prox}_{(\nu f)^*}(x)$ can be found in [95, Theorem 31.5]. The inequality $\mathrm{lip}\,(\nabla f_\nu) \leq \frac{1}{\nu}$ then follows since the proximal mapping of a closed convex function is 1-Lipschitz [95, pp. 340]. The expression (2.8) follows from rewriting $f_\nu(x) = (f^\star + \frac{\nu}{2}\|\cdot\|^2)^\star(x) = \sup_z \{\langle x, z\rangle - f^\star(z) - \frac{\nu}{2}\|z\|^2\}$ (as in e.g. [95, Theorem 16.4]) and noting that the domain of $f^\star$ is bounded in norm by L. Finally, to see that f_ν is L-Lipschitz, observe $\nabla f_\nu(x) \in \partial f(\mathrm{prox}_{\nu f}(x))$ for all x, and hence $\|\nabla f_\nu(x)\| \leq \sup\{\|v\| : y \in \mathbb{R}^d, v \in \partial f(y)\} \leq L$. □

2.3 The composite problem class

This work centers around nonsmooth and nonconvex optimization problems of the form

$$\min_x \; F(x) := g(x) + h(c(x)). \tag{2.9}$$

Throughout, we make the following assumptions on the functional components of the problem:

1. $g\colon \mathbb{R}^d \to \overline{\mathbb{R}}$ is a closed, proper, convex function;

2. $h\colon \mathbb{R}^m \to \mathbb{R}$ is a convex and L-Lipschitz continuous function:

$$|h(x) - h(y)| \leq L\|x - y\| \qquad \text{for all } x, y \in \mathbb{R}^m;$$

3. $c\colon \mathbb{R}^d \to \mathbb{R}^m$ is a C^1-smooth mapping with a β-Lipschitz continuous Jacobian map:

$$\|\nabla c(x) - \nabla c(y)\|_{\mathrm{op}} \leq \beta\|x - y\| \qquad \text{for all } x, y \in \mathbb{R}^d.$$

The values L and β will often multiply each other; hence, we define the constant $\mu := L\beta$.

2.3.1 Motivating examples

It is instructive to consider some motivating examples fitting into the framework (2.9).

Example 2.3.1 (Additive composite minimization)**.** The most prevalent example of the composite class (2.9) is additive composite minimization. In this case, the map c maps to the real line and h is the identity function:

$$\min_x \; c(x) + g(x). \tag{2.10}$$

Such problems appear often in statistical learning and imaging, for example. Numerous algorithms are available, especially when c is convex, such as proximal gradient methods and their accelerated variants [4, 86]. We will often compare and contrast techniques for general composite problems (2.9) with those specialized to this additive composite setting.

Example 2.3.2 (Nonlinear least squares)**.** The composite problem class also captures nonlinear least squares problems with bound constraints:

$$\min_x \; \|c(x)\| \qquad \text{subject to} \qquad l_i \leq x_i \leq u_i \quad \text{for } i = 1, \ldots, m.$$

Gauss-Newton type algorithm [58, 74, 76] are often the methods of choice for such problems.

Example 2.3.3 (Exact penalty formulations)**.** Consider a nonlinear optimization problem:

$$\min_x \; \{f(x) : G(x) \in \mathcal{K}\},$$

where $f\colon \mathbb{R}^d \to \mathbb{R}$ and $G\colon \mathbb{R}^d \to \mathbb{R}^m$ are smooth mappings and $\mathcal{K} \subseteq \mathbb{R}^m$ is a closed convex cone. An accompanying *penalty formulation* – ubiquitous in nonlinear optimization [35, 14, 26, 44, 17] – takes the form

$$\min_x \; f(x) + \lambda \cdot \theta_{\mathcal{K}}(G(x)),$$

where $\theta_{\mathcal{K}}\colon \mathbb{R}^m \to \mathbb{R}$ is a nonnegative convex function that is zero only on $\mathcal{K}$ and $\lambda > 0$ is a penalty parameter. For example, $\theta_{\mathcal{K}}(y)$ is often the distance of y to the convex cone $\mathcal{K}$ in some norm. This is an example of (2.9) under the identification $c(x) = (f(x), G(x))$ and $h(f, G) = f + \lambda\theta_{\mathcal{K}}(G)$.

Example 2.3.4 (Statistical estimation)**.** Often, one is interested in minimizing an error between a nonlinear process model $G(x)$ and observed data b through a misfit measure h. The resulting problem takes the form

$$\min_x \; h\big(b - G(x)\big) + g(x),$$

where g may be a convex surrogate encouraging prior structural information on x, such as the l_1-norm, squared l_2-norm, or the indicator of the nonnegative orthant. The misfit $h = \|\cdot\|_2$, in particular, appears in nonlinear least squares. The l_1-norm $h = \|\cdot\|_1$ for example is used in the Least Absolute Deviations (LAD) technique in regression [77, 105], Kalman smoothing with impulsive disturbances [3], and for robust phase retrieval [42].

Another popular class of misfit measures h is a sum $h = \sum_i h_\kappa(y_i)$ of Huber functions

$$h_\kappa(\tau) = \begin{cases} \frac{1}{2\kappa}\tau^2 & , \tau \in [-\kappa, \kappa] \\ |\tau| - \frac{\kappa}{2} & , \text{otherwise} \end{cases}$$

The Huber function figures prominently in robust regression [23, 43, 54, 68], being much less sensitive to outliers than the least squares penalty due to its linear tail growth. The function h thus defined is smooth with $\operatorname{lip}(\nabla h) \sim 1/\kappa$. Hence, in particular, the term $h(b - G(x))$ can be treated as a smooth term reducing to the setting of additive composite minimization (Example 2.3.1). On the other hand, we will see that because of the poor conditioning of the gradient ∇h, methods that take into account the non-additive composite structure can have better efficiency estimates.

Example 2.3.5 (Grey-box minimization)**.** In industrial applications, one is often interested in functions that are available only *implicitly*. For example, function and derivative evaluations may require execution of an expensive simulation. Such problems often exhibit an underlying composite structure $h(c(x))$. The penalty function h is known (and chosen) explicitly and is simple, whereas the mapping $c(x)$ and the Jacobian $\nabla c(x)$ might only be available through a simulation. Problems of this type are sometimes called *grey-box minimization problems*, in contrast to black-box minimization. The explicit separation of the hard-to-compute mapping c and the user chosen penalty h can help in designing algorithms. See for example Conn-Scheinberg-Vicente [27] and Wild [113], and references therein.

2.3.2 First-order stationary points for composite problems

Let us now explain the goal of algorithms for the problem class (2.9). Since the optimization problem (2.9) is nonconvex, it is natural to seek points x that are only first-order stationary. One makes this notion precise through subdifferentials (or generalized derivatives), which have a very explicit representation for our problem class. We recall here the relevant definitions; for more details, see for example the monographs of Mordukhovich [75] and Rockafellar-Wets [98].

Consider an arbitrary function $f\colon \mathbb{R}^d \to \overline{\mathbb{R}}$ and a point $\bar{x}$ with $f(\bar{x})$ finite. The *Fréchet subdifferential* of f at $\bar{x}$, denoted $\hat{\partial} f(\bar{x})$, is the set of all vectors v satisfying

$$f(x) \geq f(\bar{x}) + \langle v, x - \bar{x}\rangle + o(\|x - \bar{x}\|) \qquad \text{as } x \to \bar{x}.$$

Thus the inclusion $v \in \hat{\partial} f(\bar{x})$ holds precisely when the affine function $x \mapsto f(\bar{x}) + \langle v, x - \bar{x}\rangle$ underestimates f up to first-order near $\bar{x}$. In general, the limit of Fréchet subgradients $v_i \in \hat{\partial} f(x_i)$, along a sequence $x_i \to \bar{x}$, may not be a Fréchet subgradient at the limiting point $\bar{x}$. Hence, one formally enlarges the Fréchet subdifferential and defines the *limiting subdifferential* of f at $\bar{x}$, denoted $\partial f(\bar{x})$, to consist of all vectors v for which there exist sequences x_i and v_i, satisfying $v_i \in \partial f(x_i)$ and $(x_i, f(x_i), v_i) \to (\bar{x}, f(\bar{x}), v)$. We say that x is *stationary* for f if the inclusion $0 \in \partial f(x)$ holds.

For convex functions f, the subdifferentials $\hat{\partial} f(x)$ and $\partial f(x)$ coincide with the subdifferential in the sense of convex analysis (2.6), while for C^1-smooth functions f, they consist only of the gradient $\nabla f(x)$. Similarly, the situation simplifies for the composite problem class (2.9): the two subdifferentials $\hat{\partial} F$ and ∂F coincide and admit an intuitive representation through a chain-rule [98, Theorem 10.6, Corollary 10.9].

Theorem 2.3.1 (Chain rule)**.** *For the composite function F, defined in* (2.9)*, the Fréchet* *and limiting subdifferentials coincide and admit the representation*

$$\partial F(x) = \partial g(x) + \nabla c(x)^* \partial h(c(x)).$$

In summary, the algorithms we consider aim to find stationary points of F, i.e. those points x satisfying $0 \in \partial F(x)$. In "primal terms", it is worth noting that a point x is stationary for F if and only if the directional derivative of F at x is nonnegative in every direction [98, Proposition 8.32]. More precisely, the equality holds:

$$\mathrm{dist}(0; \partial F(x)) = - \inf_{v \colon \|v\| \leq 1} F'(x; v), \tag{2.11}$$

where $F'(x; v)$ is the directional derivative of F at x in direction v [98, Definition 8.1].

2.3.3 The prox-linear method

The basic algorithm we rely on for the composite problem class is the so-called prox-linear method. To motivate this scheme, let us first consider the setting of additive composite minimization (2.10). The most basic algorithm in this setting is the *proximal gradient method* [4, 86]

$$x_{k+1} := \operatorname*{argmin}_{x} \left\{ c(x_k) + \langle \nabla c(x_k), x - x_k \rangle + g(x) + \frac{1}{2t}\|x - x_k\|^2 \right\}, \tag{2.12}$$

or equivalently

$$x_{k+1} = \mathrm{prox}_{tg}\left(x_k - t\nabla c(x_k)\right).$$

Notice that an underlying assumption here is that the proximal map prox_{tg} is computable.

Convergence analysis of the prox-gradient algorithm derives from the fact that the function minimized in (2.12) is an upper model of F whenever $t \leq \beta^{-1}$. This majorization viewpoint quickly yields an algorithm for the entire problem class (2.9). The so-called *prox-linear algorithm* iteratively linearizes the map c and solves a proximal subproblem. To formalize the method, we use the following notation. For any points $z, y \in \mathbb{R}^d$ and a real $t > 0$, define

$$\begin{aligned} F(z;y) &:= g(z) + h\Big(c(y) + \nabla c(y)(z-y)\Big), \\ F_t(z;y) &:= F(z;y) + \frac{1}{2t}\left\|z-y\right\|^2, \\ S_t(y) &:= \operatorname*{argmin}_z \; F_t(z;y). \end{aligned}$$

Throughout the manuscript, we will routinely use the following estimate on the error in approximation $|F(z) - F(z;y)|$. We provide a quick proof for completeness.

Lemma 2.3.2. *For all $x, y \in \mathrm{dom}\; g$, the inequalities hold:*

$$-\frac{\mu}{2}\|z-y\|^2 \le F(z) - F(z;y) \le \frac{\mu}{2}\|z-y\|^2. \tag{2.13}$$

Proof. Since h is L-Lipschitz, we have $|F(z) - F(z;y)| \le L\big\|c(z) - \big(c(y) + \nabla c(y)(z-y)\big)\big\|$. The fundamental theorem of calculus, in turn, implies

$$\begin{aligned} \big\|c(z) - \big(c(y) + \nabla c(y)(z-y)\big)\big\| &= \left\|\int_0^1 \Big(\nabla c(y + t(z-y)) - \nabla c(y)\Big)(z-y)\,dt\right\| \\ &\le \int_0^1 \left\|\nabla c(y + t(z-y)) - \nabla c(y)\right\|_{\mathrm{op}} \left\|z-y\right\| dt \\ &\le \beta\|z-y\|^2 \left(\int_0^1 t\,dt\right) = \frac{\beta}{2}\|z-y\|^2. \end{aligned}$$

The result follows. □

In particular, Lemma 2.3.2 implies that $F_t(\cdot;y)$ is an upper model for F for any $t \le \mu^{-1}$, meaning $F_t(z;y) \ge F(z)$ for all points $y, z \in \mathrm{dom}\; g$. The *prox-linear method*, formalized in Algorithm 2, is then simply the recurrence $x_{k+1} = S_t(x_k)$. Notice that we are implicitly assuming here that the proximal subproblem (2.14) is solvable. We will discuss the impact of an inexact evaluation of $S_t(\cdot)$ in Section 2.5. Specializing to the additive composite setting (2.10), equality $S_t(x) = \mathrm{prox}_{tg}(x - t\nabla c(x))$ holds and the prox-linear method reduces to the familiar prox-gradient iteration (2.12).

The convergence rate of the prox-linear method is best stated in terms of the *prox-gradient* mapping

$$\mathcal{G}_t(x) := t^{-1}(x - S_t(x)).$$

Algorithm 2: Prox-linear method

```
Initialize : A point $x_0 \in \operatorname{dom} g$ and a real $t > 0$.
Step k: ($k \geq 0$) Compute
    $x_{k+1} = \operatorname*{argmin}_x \left\{ g(x) + h\Big(c(x_k) + \nabla c(x_k)(x - x_k)\Big) + \frac{1}{2t}\|x - x_k\|^2 \right\}.$    (2.14)
```

Observe that the optimality conditions for the proximal subproblem $\min_z F_t(z; x)$ read

$$\mathcal{G}_t(x) \in \partial g(S_t(x)) + \nabla c(x)^* \partial h(c(x) + \nabla c(x)(S_t(x) - x)).$$

In particular, for any $t > 0$, a point x is stationary for F if and only if equality $\mathcal{G}_t(x) = 0$ holds. Hence, the norm $\|\mathcal{G}_t(x)\|$ serves as a measure of "proximity to stationarity". In Section 2.4, we will establish a much more rigorous justification for why the norm $\|\mathcal{G}_t(x)\|$ provides a reliable basis for judging the quality of the point x. Let us review here the rudimentary convergence guarantees of the method in terms of the prox-gradient, as presented for example in [38, Section 5]. We provide a quick proof for completeness.

Proposition 2.3.3 (Efficiency of the pure prox-linear method)**.** *Supposing $t \leq \mu^{-1}$, the iterates generated by Algorithm 2 satisfy*

$$\min_{j=0,\ldots,N-1} \|\mathcal{G}_t(x_j)\|^2 \leq \frac{2t^{-1}\big(F(x_0) - F^*\big)}{N},$$

where we set $F^* := \lim\limits_{N \to \infty} F(x_N)$.

Proof. Taking into account that $F_t(\cdot; x_k)$ is strongly convex with modulus $1/t$, we obtain

$$F(x_k) = F_t(x_k; x_k) \geq F_t(x_{k+1}; x_k) + \tfrac{t}{2}\left\|\mathcal{G}_t(x_k)\right\|^2 \geq F(x_{k+1}) + \tfrac{t}{2}\left\|\mathcal{G}_t(x_k)\right\|^2.$$

Summing the inequalities yields

$$\min_{j=0,\ldots,N-1} \|\mathcal{G}_t(x_j)\|^2 \leq \frac{1}{N}\sum_{j=0}^{N-1} \|\mathcal{G}_t(x_j)\|^2 \leq \frac{2t^{-1}\big(F(x_0) - F^*\big)}{N},$$

as claimed. □

2.4 *Prox-gradient size $\|\mathcal{G}_t\|$ and approximate stationarity*

Before continuing the algorithmic development, let us take a closer look at what the measure $\|\mathcal{G}_t(x)\|$ tells us about "near-stationarity" of the point x. Let us first consider the additive composite setting (2.10), where the impact of the measure $\|\mathcal{G}_t(x)\|$ on near-stationarity is well-understood. As discussed on page 36, the prox-linear method reduces to the prox-gradient recurrence

$$x_{k+1} = \operatorname{prox}_{g/\beta}\left(x_k - \frac{1}{\beta}\cdot \nabla c(x_k)\right).$$

First-order optimality conditions for the proximal subproblem amounts to the inclusion

$$\mathcal{G}_{\frac{1}{\beta}}(x_k) \in \nabla c(x_k) + \partial g(x_{k+1}),$$

or equivalently

$$\mathcal{G}_{\frac{1}{\beta}}(x_k) + (\nabla c(x_{k+1}) - \nabla c(x_k)) \in \nabla c(x_{k+1}) + \partial g(x_{k+1}).$$

Notice that the right-hand-side is exactly $\partial F(x_{k+1})$. Taking into account that ∇c is β-Lipschitz, we deduce

$$\begin{aligned} \operatorname{dist}(0; \partial F(x_{k+1})) &\le \|\mathcal{G}_{\frac{1}{\beta}}(x_k)\| + \|\nabla c(x_{k+1}) - \nabla c(x_k)\| \\ &\le 2\|\mathcal{G}_{\frac{1}{\beta}}(x_k)\|. \end{aligned} \tag{2.15}$$

Thus the inequality $\|\mathcal{G}_{\frac{1}{\beta}}(x_k)\| \le \varepsilon/2$ indeed guarantees that x_{k+1} is nearly stationary for F in the sense that $\operatorname{dist}(0; \partial F(x_{k+1})) \le \varepsilon$. Taking into account (2.11), we deduce the bound on directional derivative $F'(x; u) \ge -\varepsilon$ in any unit direction u. With this in mind, the guarantee of Proposition 2.3.3 specialized to the prox-gradient method can be found for example in [86, Theorem 3].

The situation is dramatically different for the general composite class (2.10). When h is nonsmooth, the quantity $\operatorname{dist}(0; \partial F(x_{k+1}))$ will typically not even tend to zero in the limit, even though $\|\mathcal{G}_{\frac{1}{\beta}}(x_k)\|$ will tend to zero. For example, the prox-linear algorithm applied to the univariate function $f(x) = |x^2 - 1|$ and initiated at $x > 1$, will generate a decreasing sequence $x_k \to 1$ with $f'(x_k) \to 2$.[3]

Thus we must look elsewhere for an interpretation of the quantity $\|\mathcal{G}_{\frac{1}{\mu}}(x_k)\|$. We will do so by focusing on the Moreau envelope $x \mapsto F_{\frac{1}{2\mu}}(x)$ — a function that serves as a C^1-smooth approximation of F with the same stationary points. We argue in Theorem 2.4.5 that the norm of the prox-gradient $\|\mathcal{G}_{\frac{1}{\mu}}(x_k)\|$ is informative because $\|\mathcal{G}_{\frac{1}{\mu}}(x_k)\|$ is proportional to the norm of the true gradient of the Moreau envelope $\|\nabla F_{\frac{1}{2\mu}}(x)\|$. Before proving this result, we must first establish some basic properties of the Moreau envelope, which will follow from weak convexity of the composite function F; this is the content of the following section.

2.4.1 Weak convexity and the Moreau envelope of the composition

We will need the following standard definition.

Definition 2.4.1 (Weak convexity)**.** We say that a function $f\colon \mathbb{R}^d \to \overline{\mathbb{R}}$ is *ρ-weakly convex on a set U* if for any points $x, y \in U$ and $a \in [0,1]$, the approximate secant inequality holds:

$$f(ax + (1-a)y) \leq af(x) + (1-a)f(y) + \rho a(1-a)\|x - y\|^2.$$

It is well-known that for a locally Lipschitz function $f\colon \mathbb{R}^d \to \mathbb{R}$, the following are equivalent; see e.g. [30, Theorem 3.1].

1. **(Weak convexity)** f is ρ-weakly convex on $\mathbb{R}^d$.
2. **(Perturbed convexity)** The function $f + \frac{\rho}{2}\|\cdot\|^2$ is convex on $\mathbb{R}^d$.
3. **(Quadratic lower-estimators)** For any $x, y \in \mathbb{R}^d$ and $v \in \partial f(x)$, the inequality

$$f(y) \geq f(x) + \langle v, y - x\rangle - \frac{\rho}{2}\|y - x\|^2 \qquad \text{holds.}$$

[3] Notice f has three stationary points $\{-1, 0, 1\}$. Fix $y > 1$ and observe that x minimizes $f_t(\cdot; y)$ if and only if $\frac{y-x}{2ty} \in \partial|\cdot|(y^2 - 1 + 2y(x-y))$. Hence $\frac{y-x}{2ty} \cdot (y^2 - 1 + 2y(x-y)) \geq 0$. The inequality $x \leq 1$ would immediately imply a contradiction. Thus the inequality $x_0 > 1$ guarantees $x_k > 1$ for all k. The claim follows.

In particular, the following is true.

Lemma 2.4.2 (Weak convexity of the composition)**.**
The function $h \circ c$ is ρ-weakly convex on $\mathbb{R}^d$ for some $\rho \in [0, \mu]$.

Proof. To simplify notation, set $\Phi := h \circ c$. Fix two points $x, y \in \mathbb{R}^d$ and a vector $v \in \partial\Phi(x)$. We can write $v = \nabla c(x)^* w$ for some vector $w \in \partial h(c(x))$. Taking into account convexity of h and the inequality $\|c(y) - c(x) - \nabla c(x)(y - x)\| \leq \frac{\beta}{2}\|y - x\|^2$, we then deduce

$$\begin{aligned}\Phi(y) = h(c(y)) \geq h(c(x)) + \langle w, c(y) - c(x)\rangle &\geq \Phi(x) + \langle w, \nabla c(x)(y - x)\rangle - \frac{\beta\|w\|}{2}\|y - x\|^2 \\ &\geq \Phi(x) + \langle v, y - x\rangle - \frac{\mu}{2}\|y - x\|^2.\end{aligned}$$

The result follows. □

Weak convexity of F has an immediate consequence on the Moreau envelope F_ν.

Lemma 2.4.3 (Moreau envelope of the composite function)**.** *Fix $\nu \in (0, 1/\mu)$. Then the proximal map $prox_{\nu F}(x)$ is well-defined and single-valued, while the Moreau envelope F_ν is C^1-smooth with gradient*

$$\nabla F_\nu(x) = \nu^{-1}(x - prox_{\nu F}(x)). \tag{2.16}$$

Moreover, stationary points of F_ν and of F coincide.

Proof. Fix $\nu \in (0, 1/\mu)$. Lemma 2.4.2 together with [91, Theorem 4.4] immediately imply that $\text{prox}_{\nu F}(x)$ is well-defined and single-valued, while the Moreau envelope F_ν is C^1-smooth with gradient given by (2.16). Equation (2.16) then implies that x is stationary for F_ν if and only if x minimizes the function $\varphi(z) := F(z) + \frac{1}{2\nu}\|z - x\|^2$. Lemma 2.4.2 implies that φ is strongly convex, and therefore the unique minimizer z of φ is characterized by $\nu^{-1}(x - z) \in \partial F(z)$. Hence stationary points of F_ν and of F coincide. □

Thus for $\nu \in (0, 1/\mu)$, stationary points of F coincide with those of the C^1-smooth function F_ν. More useful would be to understand the impact of $\|\nabla F_\nu(x)\|$ being small, but not zero. To this end, observe the following. Lemma 2.4.3 together with the definition of the Moreau envelope implies that for any x, the point $\hat{x} := \mathrm{prox}_{\nu F}(x)$ satisfies

$$\begin{cases} \|\hat{x} - x\| & \leq \nu \|\nabla F_\nu(x)\|, \\ \mathrm{dist}(0; \partial F(\hat{x})) & \leq \|\nabla F_\nu(x)\|. \end{cases} \tag{2.17}$$

Thus a small gradient $\|\nabla F_\nu(x)\|$ implies that x is *near* a point $\hat{x}$ that is *nearly stationary* for F.

2.4.2 Prox-gradient and the gradient of the Moreau envelope

The final ingredient we need to prove Theorem 2.4.5 is the following lemma [10, Theorem 2.4.1]; we provide a short proof for completeness.

Lemma 2.4.4 (Smooth variational principle)**.** *Consider a closed function $f\colon \mathbb{R}^d \to \overline{\mathbb{R}}$ and suppose the inequality $f(x) - \inf f \leq \varepsilon$ holds for some point x and real $\varepsilon > 0$. Then for any $\lambda > 0$, the inequality holds:*

$$\|\lambda^{-1}(x - \mathit{prox}_{\lambda f}(x))\| \leq \sqrt{\frac{2\varepsilon}{\lambda}}$$

If f is α-strongly convex (possibly with $\alpha = 0$), then the estimate improves to

$$\|\lambda^{-1}(x - \mathit{prox}_{\lambda f}(x))\| \leq \sqrt{\frac{\varepsilon}{\lambda(1 + \frac{\lambda\alpha}{2})}}.$$

Proof. Fix a point $y \in \operatorname*{argmin}_z \left\{ f(z) + \frac{1}{2\lambda}\|z - x\|^2 \right\}$. We deduce

$$f(y) + \frac{1}{2\lambda}\|y - x\|^2 \leq f(x) \leq f^* + \varepsilon \leq f(y) + \varepsilon.$$

Hence we deduce $\lambda^{-1}\|y - x\| \leq \sqrt{\frac{2\varepsilon}{\lambda}}$, as claimed. If f is α-strongly convex, then the function $z \mapsto f(z) + \frac{1}{2\lambda}\|z - x\|^2$ is $(\alpha + \lambda^{-1})$-strongly convex and therefore

$$\left(f(y) + \frac{1}{2\lambda}\|y - x\|^2 \right) + \frac{\lambda^{-1} + \alpha}{2}\|y - x\|^2 \leq f(x) \leq f^* + \varepsilon \leq f(y) + \varepsilon.$$

The claimed inequality follows along the same lines. □

We can now quantify the precise relationship between the norm of the prox-gradient $\|\mathcal{G}_t(x)\|$ and the norm of the true gradient of the Moreau envelope $\|\nabla F_{\frac{t}{1+t\mu}}(x)\|$.

Theorem 2.4.5 (Prox-gradient and near-stationarity). *For any point x and real constant $t > 0$, the inequality holds:*

$$\tfrac{1}{(1+\mu t)(1+\sqrt{\mu t})} \left\|\nabla F_{\frac{t}{1+t\mu}}(x)\right\| \leq \|\mathcal{G}_t(x)\| \leq \tfrac{1+2t\mu}{1+t\mu}\left(\sqrt{\tfrac{t\mu}{1+t\mu}}+1\right)\left\|\nabla F_{\frac{t}{1+t\mu}}(x)\right\|. \tag{2.18}$$

Proof. To simplify notation, throughout the proof set

$$\begin{aligned}\bar{x} &:= S_t(x) = \operatorname*{argmin}_z F_t(z;x),\\ \hat{x} &:= \operatorname{prox}_{\frac{tF}{1+t\mu}}(x) = \operatorname*{argmin}_z \ \{F(z) + \tfrac{\mu+t^{-1}}{2}\|z-x\|^2\}.\end{aligned}$$

Notice that $\hat{x}$ is well-defined by Lemma 2.4.3.

We begin by establishing the first inequality in (2.18). For any point z, we successively deduce

$$\begin{aligned}F(z) &\geq F_t(z;x) - \tfrac{\mu+t^{-1}}{2}\|z-x\|^2 \geq F_t(\bar{x};x) + \tfrac{1}{2t}\|\bar{x}-z\|^2 - \tfrac{\mu+t^{-1}}{2}\|z-x\|^2\\ &\geq F(\bar{x}) + \frac{1}{2t}\|\bar{x}-z\|^2 - \tfrac{\mu+t^{-1}}{2}\|z-x\|^2 + \tfrac{t^{-1}-\mu}{2}\|\bar{x}-x\|^2,\end{aligned} \tag{2.19}$$

where the first and third inequalities follow from (2.13) and the second from strong convexity of $F_t(\cdot;x)$.

Define the function $\zeta(z) := F(z) + \frac{\mu+t^{-1}}{2}\|z-x\|^2 - \frac{1}{2t}\|\bar{x}-z\|^2$ and notice that ζ is convex by Lemma 2.4.2. Inequality (2.19) directly implies

$$\zeta(\bar{x}) - \inf\zeta \leq \left(F(\bar{x}) + \tfrac{\mu+t^{-1}}{2}\|\bar{x}-x\|^2\right) - \left(F(\bar{x}) + \tfrac{t^{-1}-\mu}{2}\|\bar{x}-x\|^2\right) = \mu\|\bar{x}-x\|^2.$$

Notice the relation, $\operatorname{prox}_{t\zeta}(\bar{x}) = \operatorname{prox}_{\frac{tF}{1+t\mu}}(x) = \hat{x}$. Setting $\lambda := t$ and $\varepsilon := \mu\|\bar{x}-x\|^2$ and using Lemma 2.4.4 (convex case $\alpha = 0$) with $\bar{x}$ in place of x, we conclude

$$\sqrt{\tfrac{\mu}{t}}\|\bar{x}-x\| \geq \|t^{-1}(\bar{x}-\operatorname{prox}_{t\zeta}(\bar{x}))\| = \|t^{-1}(\bar{x}-\hat{x})\| \geq \|t^{-1}(x-\hat{x})\| - \|t^{-1}(\bar{x}-x)\|.$$

Rearranging and using (2.16) yields the first inequality in (2.18), as claimed.

We next establish the second inequality in (2.18). The argument is in the same spirit as the previous part of the proof. For any point z, we successively deduce

$$\begin{aligned} F_t(z;x) &\geq (F(z) + \tfrac{\mu+t^{-1}}{2}\|z-x\|^2) - \mu\|z-x\|^2 \\ &\geq F(\hat{x}) + \tfrac{\mu+t^{-1}}{2}\|\hat{x}-x\|^2 + \tfrac{1}{2t}\|\hat{x}-z\|^2 - \mu\|z-x\|^2, \end{aligned} \tag{2.20}$$

where the first inequality follows from (2.13) and the second from t^{-1}-strong convexity of $z \mapsto F(z) + \frac{\mu+t^{-1}}{2}\|z-x\|^2$. Define now the function

$$\Psi(z) := F_t(z;x) - \tfrac{1}{2t}\|\hat{x}-z\|^2 + \mu\|z-x\|^2.$$

Combining (2.13) and (2.20), we deduce

$$\Psi(\hat{x}) - \inf \Psi \leq \Big(F_t(\hat{x};x) + \mu\|\hat{x}-x\|^2\Big) - \Big(F(\hat{x}) + \tfrac{\mu+t^{-1}}{2}\|\hat{x}-x\|^2\Big) \leq \mu\|\hat{x}-x\|^2.$$

Notice that Ψ is strongly convex with parameter $\alpha := 2\mu$. Setting $\varepsilon := \mu\|\hat{x}-x\|^2$ and $\lambda = t$, and applying Lemma 2.4.4 with $\hat{x}$ in place of x, we deduce

$$\sqrt{\tfrac{\mu}{t(1+t\mu)}}\|\hat{x}-x\| \geq \|t^{-1}(\hat{x} - \mathrm{prox}_{t\Psi}(\hat{x}))\| \geq \|t^{-1}(x - \mathrm{prox}_{t\Psi}(\hat{x}))\| - \|t^{-1}(\hat{x}-x)\|. \tag{2.21}$$

To simplify notation, set $\hat{z} := \mathrm{prox}_{t\Psi}(\hat{x})$. By definition of Ψ, equality

$$\hat{z} = \operatorname*{argmin}_z \;\; \left\{F_t(z;x) + \mu\|z-x\|^2\right\} \qquad \text{holds,}$$

and therefore $2\mu(x-\hat{z}) \in \partial F_t(\hat{z};x)$. Taking into account that $F_t(\cdot;x)$ is t^{-1}-strongly convex, we deduce

$$\|2\mu(x-\hat{z})\| \geq \mathrm{dist}\,(0;\partial F_t(\hat{z};x)) \geq t^{-1}\|\hat{z}-\bar{x}\| \geq \|t^{-1}(x-\bar{x})\| - \|t^{-1}(x-\hat{z})\|.$$

Rearranging and combining the estimate with (2.16), (2.21) yields the second inequality in (2.18). □

In the most important setting $t = 1/\mu$, Theorem 2.4.5 reduces to the estimate

$$\tfrac{1}{4}\left\|\nabla F_{\frac{1}{2\mu}}(x)\right\| \leq \|\mathcal{G}_{1/\mu}(x)\| \leq \tfrac{3}{2}\left(1+\tfrac{1}{\sqrt{2}}\right)\left\|\nabla F_{\frac{1}{2\mu}}(x)\right\|. \tag{2.22}$$

A closely related result has recently appeared in [38, Theorem 5.3], with a different proof, and has been extended to a more general class of Taylor-like approximations in [36]. Combining (2.22) and (2.17) we deduce that for any point x, there exists a point $\hat{x}$ (namely $\hat{x} = \mathrm{prox}_{F/2\mu}(x)))$ satisfying

$$\begin{cases} \|\hat{x} - x\| & \leq \frac{2}{\mu}\|\mathcal{G}_{1/\mu}(x)\|, \\ \\ \mathrm{dist}(0; \partial F(\hat{x})) & \leq 4\|\mathcal{G}_{1/\mu}(x)\|. \end{cases} \tag{2.23}$$

Thus if $\|\mathcal{G}_{1/\mu}(x)\|$ is small, the point x is "near" some point $\hat{x}$ that is "nearly-stationary" for F. Notice that $\hat{x}$ is not computable, since it requires evaluation of $\mathrm{prox}_{F/2\mu}$. Computing $\hat{x}$ is not the point, however; the sole purpose of $\hat{x}$ is to certify that x is approximately stationary in the sense of (2.23).

2.5 Inexact analysis of the prox-linear method

In practice, it is often impossible to solve the proximal subproblems $\min_z F_t(z; y)$ exactly. In this section, we explain the effect of inexactness in the proximal subproblems (2.14) on the overall performance of the prox-linear algorithm. By "inexactness", one can mean a variety of concepts. Two most natural ones are that of (i) terminating the subproblems based on near-optimality in function value and (ii) terminating based on "near-stationarity".

Which of the two criteria is used depends on the algorithms that are available for solving the proximal subproblems. If primal-dual interior-point methods are applicable, then termination based on near-optimality in function value is most appropriate. When the subproblems themselves can only be solved by first-order methods, the situation is less clear. In particular, if near-optimality in function value is the goal, then one must use saddle-point methods. Efficiency estimates of saddle-point algorithms, on the other hand, depend on the diameter of the feasible region, rather than on the quality of the initial iterate (e.g. distance of initial iterate to the optimal solution). Thus saddle-point methods cannot be directly warm-started, that is one cannot easily use iterates from previous prox-linear subproblems to speed up the algorithm for the current subproblem. Moreover, there is a conceptual incompatibility of the prox-linear method with termination based on functional near-optimality. Indeed, the prox-linear method seeks to make the stationarity measure $\|\mathcal{G}_t(x)\|$ small, and so it seems more fitting that the proximal subproblems are solved based on near-stationarity themselves. In this section, we consider both termination criteria. The arguments are quick modifications of the proof of Proposition 2.3.3.

2.5.1 Near-optimality in the subproblems

We first consider the effect of solving the proximal subproblems up to a tolerance on function values. Given a tolerance $\varepsilon > 0$, we say that a point x is an *ε-approximate minimizer* of a function $f\colon \mathbb{R}^d \to \overline{\mathbb{R}}$ whenever the inequality holds:

$$f(x) \leq \inf\, f + \varepsilon.$$

Consider now a sequence of tolerances $\varepsilon_k \geq 0$ for $k = 1, 2 \ldots, \infty$. Then given a current iterate x_k, an *inexact prox-linear algorithm* for minimizing F can simply declare x_{k+1} to be an ε_{k+1}-approximate minimizer of $F_t(\cdot; x_k)$. We record this scheme in Algorithm 3.

```
Algorithm 3: Inexact prox-linear method: near-optimality
  Initialize : A point $x_0 \in \operatorname{dom} g$, a real $t > 0$, and a sequence $\{\varepsilon_i\}_{i=1}^{\infty} \subset [0, +\infty)$.
  Step k: ($k \geq 0$) Set $x_{k+1}$ to be an $\varepsilon_{k+1}$-approximate minimizer of $F_t(\cdot; x_k)$.
```

Before stating convergence guarantees of the method, we record the following observation stating that the step-size of the inexact prox-linear method $\|x_{k+1} - x_k\|$ and the accuracy ε_k jointly control the size of the true prox-gradient $\|\mathcal{G}_t(x_k)\|$. As a consequence, the step-sizes $\|x_{k+1} - x_k\|$ generated throughout the algorithm can be used as surrogates for the true stationarity measure $\|\mathcal{G}_t(x_k)\|$.

Lemma 2.5.1. *Suppose x^+ is an ε-approximate minimizer of $F_t(\cdot; x)$. Then the inequality holds:*

$$\|\mathcal{G}_t(x)\|^2 \leq 4t^{-1}\varepsilon + 2\left\|t^{-1}(x^+ - x)\right\|^2.$$

Proof. Let z^* be the true minimizer of $F_t(\cdot; x)$. We successively deduce

$$\begin{aligned} \|\mathcal{G}_t(x)\|^2 &\leq \frac{4}{t} \cdot \frac{1}{2t} \left\|x^+ - z^*\right\|^2 + 2 \left\|t^{-1}(x^+ - x)\right\|^2 \\ &\leq \frac{4}{t} \cdot \big(F_t(x^+; x) - F_t(z^*; x)\big) + 2 \left\|t^{-1}(x^+ - x)\right\|^2 \\ &\leq \frac{4}{t} \cdot \varepsilon + 2 \left\|t^{-1}(x^+ - x)\right\|^2, \end{aligned} \tag{2.24}$$

where the first inequality follows from the triangle inequality and the estimate $(a+b)^2 \leq 2(a^2 + b^2)$ for any reals a, b, and the second inequality is an immediate consequence of strong convexity of the function $F_t(\cdot; x)$. □

The inexact prox-linear algorithm comes equipped with the following guarantee.

Theorem 2.5.2 (Convergence of the inexact prox-linear algorithm: near-optimality)**.** *Supposing $t \leq \mu^{-1}$, the iterates generated by Algorithm 3 satisfy*

$$\min_{j=0,\ldots,N-1} \|\mathcal{G}_t(x_j)\|^2 \leq \frac{2t^{-1}\big(F(x_0) - F^* + \sum_{j=1}^{N} \varepsilon_j\big)}{N},$$

where we set $F^* := \liminf_{k\to\infty} F(x_k)$.

Proof. Let x_k^* be the exact minimizer of $F_t(\cdot; x_k)$. Note then the equality $\mathcal{G}_t(x_k) = t^{-1}(x_k^* - x_k)$. Taking into account that $F_t(\cdot; x_k)$ is strongly convex with modulus $1/t$, we deduce

$$F(x_k) = F_t(x_k; x_k) \geq F_t(x_k^*; x_k) + \tfrac{t}{2} \|\mathcal{G}_t(x_k)\|^2 \geq F_t(x_{k+1}; x_k) - \varepsilon_{k+1} + \tfrac{t}{2} \|\mathcal{G}_t(x_k)\|^2.$$

Then the inequality $t \leq \mu^{-1}$ along with (2.13) implies that $F_t(\cdot; x_k)$ is an upper model of $F(\cdot)$ and therefore

$$F(x_k) \geq F(x_{k+1}) - \varepsilon_{k+1} + \tfrac{t}{2} \|\mathcal{G}_t(x_k)\|^2. \tag{2.25}$$

We conclude

$$\begin{aligned} \min_{j=0,\ldots,N-1} \|\mathcal{G}_t(x_j)\|^2 \leq \frac{1}{N} \sum_{j=0}^{N-1} \|\mathcal{G}_t(x_j)\|^2 &\leq \frac{2t^{-1}\left(\sum_{j=0}^{N-1} F(x_j) - F(x_{j+1}) + \sum_{j=0}^{N-1} \varepsilon_{j+1}\right)}{N} \\ &\leq \frac{2t^{-1}\big(F(x_0) - F^* + \sum_{j=0}^{N-1} \varepsilon_{j+1}\big)}{N}. \end{aligned}$$

The proof is complete. □

Thus in order to maintain the rate afforded by the exact prox-linear method, it suffices for the errors $\{\varepsilon_k\}_{k=1}^{\infty}$ to be summable; e.g. set $\varepsilon_k \sim \frac{1}{k^{1+q}}$ with $q > 0$.

2.5.2 Near-stationarity in the subproblems

In the previous section, we considered the effect of solving the proximal subproblems up to an accuracy in functional error. We now consider instead a model of inexactness for the proximal subproblems based on near-stationarity. A first naive attempt would be to consider a point z to be ε-stationary for the proximal subproblem, $\min F_t(\cdot; x)$, if it satisfies

$$\operatorname{dist}(0; \partial_z F_t(z; x)) \leq \varepsilon.$$

This assumption, however, is not reasonable since first-order methods for this problem do not produce such points z, unless h is smooth. Instead, let us look at the Fenchel dual problem. To simplify notation, write the target subproblem $\min F_t(\cdot; x)$ as

$$\min_z \; h(b - Az) + G(z) \tag{2.26}$$

under the identification $G(z) = g(z) + \frac{1}{2t}\|z - x\|^2$, $A = -\nabla c(x)$, and $b = c(x) - \nabla c(x)x$. Notice that G is t^{-1}-strongly convex and therefore $G^\star$ is C^1-smooth with t-Lipschitz gradient. The Fenchel dual problem, after negation, takes the form [98, Example 11.41]:

$$\min_w \; \varphi(w) := G^\star(A^* w) - \langle b, w\rangle + h^\star(w). \tag{2.27}$$

Thus the dual objective function φ is a sum of a smooth convex function $G^\star(A^*w) - \langle b, w\rangle$ and the simple nonsmooth convex term $h^\star$. Later on, when x depends on an iteration counter k, we will use the notation φ_k, G_k, A_k, b_k instead to make precise that these objects depend on k.

Typical first-order methods, such as prox-gradient and its accelerated variants can generate a point w for the problem (2.27) satisfying

$$\operatorname{dist}(0; \partial\varphi(w)) \leq \varepsilon \tag{2.28}$$

up to any specified tolerance $\varepsilon > 0$. Such schemes in each iteration only require evaluation of the gradient of the smooth function $G^\star(A^*w) - \langle b, w\rangle$ along with knowledge of a Lipschitz constant of the gradient, and evaluation of the proximal map of $h^\star$. For ease of reference, we record these quantities here in terms of the original functional components of the composite problem (2.9). Since the proof is standard, we have placed it in Appendix A.1.

Lemma 2.5.3. *The following are true for all points z and w and real $t > 0$:*

- *The equation holds:*
$$prox_{th^\star}(w) = t\left(w/t - prox_{h/t}(w/t)\right). \tag{2.29}$$
- *The equations hold:*
$$G^\star(z) = (g^\star)_{1/t}(z + x/t) - \tfrac{1}{2t}\|x\|^2 \qquad \textit{and} \qquad \nabla G^\star(z) = prox_{tg}(x + tz). \tag{2.30}$$
Consequently, the gradient map $\nabla\Big(G^\star \circ A^* - \langle \cdot, b\rangle\Big)$ is Lipschitz continuous with constant $t\|\nabla c(x)\|_{op}^2$ and admits the representation:
$$\nabla\Big(G^\star \circ A^* - \langle b, \cdot\rangle\Big)(w) = \nabla c(x)\Big(x + prox_{tg}(x - t\nabla c(x)^* w)\Big) - c(x). \tag{2.31}$$

Thus, suppose we have found a point w satisfying (2.28). How can we then generate a primal iterate x^+ at which to form the prox-linear subproblem for the next step? The following lemma provides a simple recipe for doing exactly that. It shows how to generate from w a point that is a true minimizer to a slight perturbation of the proximal subproblem.

Lemma 2.5.4 (Primal recovery from dual ε-stationarity)**.** *Let φ be the function defined in* (2.27)*. Fix a point $w \in \operatorname{dom}\varphi$ and a vector $\zeta \in \partial\varphi(w)$. Then the point $x := \nabla G^\star(A^*w)$ is the true minimizer of the problem*
$$\min_z \; h(\zeta + b - Az) + G(z). \tag{2.32}$$

Proof. Appealing to the chain rule, $\partial\varphi(w) = A\nabla G^\star(A^*w) - b + \partial h^\star(w)$, we deduce
$$\zeta + b \in A\nabla G^\star(A^*w) + \partial h^\star(w) = Ax + \partial h^\star(w).$$

The relation (2.7) then implies $w \in \partial h(\zeta+b-Ax)$. Applying A^* to both sides and rearranging yields

$$0 \in -A^*\partial h(\zeta + b - Ax) + A^* w \subseteq -A^*\partial h(\zeta + b - Ax) + \partial G(x),$$

where the last inclusion follows from applying (2.7) to G. The right-hand-side is exactly the subdifferential of the objective function in (2.32) evaluated at x. The result follows. □

This lemma directly motivates the following inexact extension of the prox-linear algorithm (Algorithm 4), based on dual near-stationary points.

```
Algorithm 4: Inexact prox-linear method: near-stationarity
    Initialize : A point $x_0 \in \operatorname{dom} g$, a real $t > 0$, and a sequence $\{\varepsilon_i\}_{i=1}^{\infty} \subset [0, +\infty)$.
    Step k: ($k \geq 0$) Find $(x_{k+1}, \zeta_{k+1})$ such that $\|\zeta_{k+1}\| \leq \varepsilon_{k+1}$ and $x_{k+1}$ is the minimizer of the function
        $z \mapsto g(z) + h\Big(\zeta_{k+1} + c(x_k) + \nabla c(x_k)(z - x_k)\Big) + \frac{1}{2t}\|z - x_k\|^2.$     (2.33)
```

Algorithm 4 is stated in a way most useful for convergence analysis. On the other hand, it is not very explicit. To crystallize the ideas, let us concretely describe how one can implement step k of the scheme. First, we find a point w_{k+1} that is ε_{k+1}-stationary for the dual problem (2.27). More precisely, we find a pair (w_{k+1}, ζ_{k+1}) satisfying $\zeta_{k+1} \in \partial \varphi_k(w_{k+1})$ and $\|\zeta_{k+1}\| \leq \varepsilon_{k+1}$. We can achieve this by a proximal gradient method (or its accelerated variants) on the dual problem (2.27). Then combining Lemma 2.5.4 with equation (2.30), we conclude that we can simply set

$$x_{k+1} := \nabla G^{\star}(A^* w_{k+1}) = \operatorname{prox}_{tg}(x_k - t\nabla c(x_k)^* w_{k+1}).$$

We record this more explicit description of Algorithm 4 in Algorithm 5. The reader should keep in mind that even though Algorithm 5 is more explicit, the convergence analysis we present will use the description in Algorithm 4.

```
Algorithm 5: Inexact prox-linear method: near-stationarity (explicit)
  Initialize : A point $x_0 \in \operatorname{dom} g$, a real $t > 0$, and a sequence $\{\varepsilon_i\}_{i=1}^{\infty} \subset [0, +\infty)$.
  Step k: ($k \geq 0$) Define the function
      $$\varphi_k(w) := (g^\star)_{1/t}\Big(x_k/t - \nabla c(x_k)^* w\Big) - \big\langle c(x_k) - \nabla c(x_k)x_k, w\big\rangle + h^\star(w).$$
  Find a point $w_{k+1}$ satisfying $\operatorname{dist}(0; \partial\varphi_k(w_{k+1})) \leq \varepsilon_{k+1}$.
  Set $x_{k+1} = \operatorname{prox}_{tg}(x_k - t\nabla c(x_k)^* w_{k+1})$.
```

Before stating convergence guarantees of the method, we record the following observation stating that the step-size $\|x_{k+1} - x_k\|$ and the error ε_{k+1} jointly control the stationarity measure $\|\mathcal{G}_t(x_k)\|$. In other words, one can use the step-size $\|x_{k+1} - x_k\|$, generated throughout the algorithm, as a surrogate for the true stationarity measure $\|\mathcal{G}_t(x_k)\|$.

Lemma 2.5.5. *Suppose x^+ is a minimizer of the function*

$$z \mapsto g(z) + h\Big(\zeta + c(x) + \nabla c(x)(z - x)\Big) + \frac{1}{2t}\|z - x\|^2$$

for some vector ζ. Then for any real $t > 0$, the inequality holds:

$$\|\mathcal{G}_t(x)\|^2 \leq 8Lt^{-1} \cdot \|\zeta\| + 2\left\|t^{-1}(x^+ - x)\right\|^2. \tag{2.34}$$

Proof. Define the function

$$l(z) = g(z) + h\Big(\zeta + c(x) + \nabla c(x)(z - x)\Big) + \frac{1}{2t}\|z - x\|^2.$$

Let z^* be the true minimizer of $F_t(\cdot;x)$. We successively deduce

$$\begin{aligned}
\|\mathcal{G}_t(x)\|^2 &\leq \frac{4}{t}\cdot\frac{1}{2t}\left\|x^+ - z^*\right\|^2 + 2\left\|t^{-1}(x^+ - x)\right\|^2 \\
&\leq \frac{4}{t}\cdot\big(F_t(x^+;x) - F_t(z^*;x)\big) + 2\left\|t^{-1}(x^+ - x)\right\|^2 \\
&\leq \frac{4}{t}(l(x^+) - l(z^*) + 2L\|\zeta\|) + 2\left\|t^{-1}(x^+ - x)\right\|^2 \\
&\leq 8t^{-1}L\|\zeta\| + 2\left\|t^{-1}(x^+ - x)\right\|^2,
\end{aligned} \tag{2.35}$$

where the first inequality follows from the triangle inequality and the estimate $(a+b)^2 \leq 2(a^2+b^2)$ for any reals a,b, the second inequality is an immediate consequence of strong convexity of the function $F_t(\cdot;x)$, and the third follows from Lipschitz continuity of h. □

Theorem 2.5.6 explains the convergence guarantees of the method; c.f. Proposition 2.3.3.

Theorem 2.5.6 (Convergence of the inexact prox-linear method: near-stationarity)**.** *Supposing $t \leq \mu^{-1}$, the iterates generated by Algorithm 4 satisfy*

$$\min_{j=0,\ldots,N-1} \|\mathcal{G}_t(x_j)\|^2 \leq \frac{4t^{-1}\big(F(x_0) - F^* + 4L\cdot\sum_{j=1}^{N}\varepsilon_j\big)}{N},$$

where we set $F^* := \liminf_{k\to\infty} F(x_k)$.

Proof. Observe the inequalities:

$$\begin{aligned}
F(x_{k+1}) &\leq F_t(x_{k+1};x_k) \\
&\qquad \leq h\big(\zeta_{k+1} + c(x_k) + \nabla c(x_k)(x_{k+1} - x_k)\big) + g(x_{k+1}) + \tfrac{1}{2t}\left\|x_{k+1} - x_k\right\|^2 + L\cdot\varepsilon_{k+1}.
\end{aligned}$$

Since the point x_{k+1} minimizes the $\frac{1}{t}$-strongly convex function in (2.33), we deduce

$$\begin{aligned}
F(x_{k+1}) &\leq h\big(\zeta_{k+1} + c(x_k)\big) + g(x_k) + L\cdot\varepsilon_{k+1} - \tfrac{1}{2t}\left\|x_{k+1} - x_k\right\|^2 \\
&\leq F(x_k) + 2L\cdot\varepsilon_{k+1} - \tfrac{1}{2t}\left\|x_{k+1} - x_k\right\|^2.
\end{aligned} \tag{2.36}$$

Summing along the indices $j = 0,\ldots,N-1$ yields

$$\sum_{j=0}^{N-1} \|t^{-1}(x_{j+1} - x_j)\|^2 \leq \frac{2}{t}\left(F(x_0) - F^* + 2L\sum_{j=0}^{N-1}\varepsilon_{j+1}\right).$$

Taking into account Lemma 2.5.5, we deduce

$$\min_{j=0,1,\ldots,N-1} \|\mathcal{G}_t(x_j)\|^2 \leq \frac{1}{N}\sum_{j=0}^{N-1} \|\mathcal{G}_t(x_j)\|^2 \leq \frac{4t^{-1}(F(x_0) - F^* + 4L\sum_{j=1}^{N}\varepsilon_j)}{N}, \tag{2.37}$$

as claimed. □

In particular, to maintain the same rate in N as the exact prox-linear method in Proposition 2.3.3, we must be sure that the sequence ε_k is summable. Hence, we can set $\varepsilon_k \sim \frac{1}{k^{1+q}}$ for any $q > 0$.

2.6 Overall complexity for the composite problem class

In light of the results of Section 2.5, we can now use the inexact prox-linear method to derive efficiency estimates for the composite problem class (2.9), where the proximal subproblems are themselves solved by first-order methods. As is standard, we will assume that the functions h and g are *prox-friendly*, meaning that prox_{th} and prox_{tg} can be evaluated. Given a target accuracy $\varepsilon > 0$, we aim to determine the number of *basic operations* – matrix-vector multiplications $\nabla c(x)v$ and $\nabla c(x)^*w$, evaluations of prox_{th}, prox_{tg} – needed to find a point x satisfying $\|\mathcal{G}_t(x)\| \leq \varepsilon$. To make progress, in this section we also assume that we have available a real value, denoted $\|\nabla c\|$, satisfying

$$\|\nabla c\| \geq \sup_{x \in \text{dom } g} \|\nabla c(x)\|_{\text{op}}.$$

In particular, we assume that the right-hand-side is finite. Strictly speaking, we only need the inequality $\|\nabla c\| \geq \|\nabla c(x_k)\|_{\text{op}}$ to holds along an iterate sequence x_k generated by the inexact prox-linear method. This assumption is completely expected: even when c is a linear map, convergence rates of first-order methods for the composite problem (2.9) depend on some norm of the Jacobian ∇c.

The strategy we propose can be succinctly summarized as follows:

- (Smoothing+prox-linear+fast-gradient) We will replace h by a smooth approximation (Moreau envelope), with a careful choice of the smoothing parameter. Then we will apply an inexact prox-linear method to the smoothed problem, with the proximal subproblems approximately solved by fast-gradient methods.

The basis for the ensuing analysis is the fast-gradient method of Nesterov [86] for minimizing convex additive composite problems. The following section recalls the scheme and records its efficiency guarantees, for ease of reference.

2.6.1 Interlude: fast gradient method for additive convex composite problems

This section discusses a scheme from [86] that can be applied to any problem of the form

$$\min_x \; f^p(x) := f(x) + p(x), \tag{2.38}$$

where $f\colon \mathbb{R}^d \to \mathbb{R}$ is a convex C^1-smooth function with L_f-Lipschitz gradient and $p\colon \mathbb{R}^d \to \overline{\mathbb{R}}$ is a closed α-strongly convex function ($\alpha \geq 0$). The setting $\alpha = 0$ signifies that p is just convex.

We record in Algorithm 6 the so-called "fast-gradient method" for such problems [86, Accelerated Method].

The method comes equipped with the following guarantee [86, Theorem 6].

Theorem 2.6.1. *Let x^* be a minimizer of f^p and suppose $\alpha > 0$. Then the iterates x_j generated by Algorithm 6 satisfy:*

$$f^p(x_j) - f^p(x^*) \leq \left(1 + \sqrt{\frac{\alpha}{2L_f}}\right)^{-2(j-1)} \frac{L_f}{4}\|x^* - x_0\|^2.$$

Let us now make a few observations that we will call on shortly. First, each iteration of Algorithm 6 only requires two gradient computations, $\nabla f(y_j)$ in (2.40) and $\nabla f(x_{j+1})$ in (2.41), and two proximal operations, prox_{p/L_f} in (2.40) and prox_p in (2.39).

Secondly, let us translate the estimates in Theorem 2.6.1 to estimates based on desired accuracy. Namely, simple arithmetic shows that the inequality

$$f^p(x_j) - f^p(x^*) \leq \varepsilon$$

```
Algorithm 6: Fast gradient method of Nesterov [86, Accelerated Method]
  Initialize : Fix a point $x_0 \in \operatorname{dom} p$, set $\theta_0 = 0$, define the function
      $\psi_0(x) := \frac{1}{2}\|x - x_0\|^2$.
  Step j: ($j \geq 0$) Find $a_{j+1} > 0$ from the equation
      $\frac{a_{j+1}^2}{\theta_j + a_{j+1}} = 2\frac{1+\alpha\theta_j}{L_f}$.
    Compute the following:
      $\theta_{j+1} = \theta_j + a_{j+1},$
      $v_j = \operatorname*{argmin}_x \; \psi_j(x),$    (2.39)
      $y_j = \frac{\theta_j x_j + a_{j+1} v_j}{\theta_{j+1}},$
      $x_{j+1} = \operatorname*{argmin}_x \; \{f(y_j) + \langle \nabla f(y_j), x - y_j \rangle + \frac{L_f}{2}\|x - y_j\|^2 + p(x)\}.$    (2.40)
    Define the function
      $\psi_{j+1}(x) = \psi_j(x) + a_{j+1}[f(x_{j+1}) + \langle \nabla f(x_{j+1}), x - x_{j+1} \rangle + p(x)].$    (2.41)
```

holds as soon as the number of iterations j satisfies

$$j \geq 1 + \sqrt{\frac{L_f}{2\alpha}} \cdot \log\left(\frac{L_f\|x^* - x_0\|^2}{4\varepsilon}\right). \tag{2.42}$$

Finally, given a point x consider a single prox-gradient iteration $\hat{x} := \operatorname{prox}_{\frac{p}{L_f}}\left(x - \frac{1}{L_f}\nabla f(x)\right)$. Then we successively deduce

$$\operatorname{dist}^2(0; \partial f^p(\hat{x})) \leq 4\|L_f(\hat{x} - x)\|^2 \leq 8L_f(f^p(x) - f^p(\hat{x})) \leq 8L_f(f^p(x) - f^p(x^*))$$

where the first inequality is (2.15) and the second is the descent guarantee of the prox-gradient method (e.g. [86, Theorem 1]). Thus the inequality $f^p(x) - f^p(x^*) \leq \varepsilon^2/(8L_f)$ would immediately imply $\operatorname{dist}(0; \partial f^p(\hat{x})) \leq \varepsilon$. Therefore, let us add an extra prox-gradient step $\hat{x}_j := \mathrm{prox}_{\frac{p}{L_f}}\left(x_j - \frac{1}{L_f}\nabla f(x_j)\right)$ to each iteration of Algorithm 6. Appealing to the linear rate in (2.42), we then deduce that we can be sure of the inequality

$$\mathrm{dist}(0; \partial f^p(\hat{x}_j)) \le \varepsilon$$

as soon as the number of iterations j satisfies

$$j \ge 1 + \sqrt{\frac{L_f}{2\alpha}} \cdot \log\left(\frac{2L_f^2\|x^* - x_0\|^2}{\varepsilon^2}\right). \tag{2.43}$$

With this modification, each iteration of the scheme requires two gradient evaluations of f and three proximal operations of p.

2.6.2 Total cost if h is smooth

In this section, we will assume that h is already C^1-smooth with the gradient having Lipschitz constant L_h, and calculate the overall cost of the inexact prox-linear method that wraps a linearly convergent method for the proximal subproblems. As we have discussed in Section 2.5, the proximal subproblems can either be approximately solved by primal methods or by dual methods. The dual methods are better adapted for a global analysis, since the dual problem has a bounded domain; therefore let us look first at that setting.

Remark 1 (Asymptotic notation)**.** To make clear dependence on the problem's data, we will sometimes use asymptotic notation [11, Section 3.5]. For two functions ψ and Ψ of a vector $\omega \in \mathbb{R}^\ell$, the symbol $\psi(\omega) = \mathcal{O}(\Psi(\omega))$ will mean that there exist constants $K, C > 0$ such that the inequality, $|\psi(\omega)| \le C \cdot |\Psi(\omega)|$, holds for all ω satisfying $\omega_i \ge K$ for all $i = 1, \ldots, \ell$. When using asymptotic notation in this section, we will use the vector ω to encode the data of the problem $\omega = (\|\nabla c\|, L_h, L, \beta, F(x_0) - F^*, 1/\varepsilon)$. In the seeing h is not differentiable, L_h will be omitted from ω.

Total cost based on dual near-stationarity in the subproblems

We consider the near-stationarity model of inexactness as in Section 2.5.2. Namely, let us compute the total cost of Algorithm 5, when each subproblem $\min_w \varphi_k(w)$ is approximately minimized by the fast-gradient method (Algorithm 6). In the notation of Section 2.6.1, we set $f(w) = G_k^\star(A_k^* w) - \langle b_k, w\rangle$ and $p = h^\star$. By Lemma 2.5.3, the function f is C^1-smooth with gradient having Lipschitz constant $L_f := t\|\nabla c(x_k)\|_{\mathrm{op}}^2$. Since ∇h is assumed to be L_h-Lipschitz, we deduce that $h^\star$ is $\frac{1}{L_h}$-strongly convex. Notice moreover that since h is L-Lipschitz, any point in dom $h^\star$ is bounded in norm by L; hence the diameter of dom $h^\star$ is at most $2L$. Let us now apply Algorithm 6 (with the extra prox-gradient step) to the problem $\min_w \varphi_k(w) = f(w) + p(w)$. According to the estimate (2.43), we will be able to find the desired point w_{k+1} satisfying $\mathrm{dist}(0; \partial\varphi_k(w_{k+1}))) \leq \varepsilon_{k+1}$ after at most

$$1 + \left\lceil \sqrt{\frac{t\|\nabla c(x_k)\|_{\mathrm{op}}^2 L_h}{2}} \cdot \log\left(\frac{8t^2\|\nabla c(x_k)\|_{\mathrm{op}}^4 L^2}{\varepsilon_{k+1}^2}\right)\right\rceil. \tag{2.44}$$

iterations of the fast-gradient method. According to Lemma 2.5.3, each gradient evaluation ∇f requires two-matrix vector multiplications and one proximal operation of g, while the proximal operation of p amounts to a single proximal operation of h. Thus each iteration of Algorithm 6, with the extra prox-gradient step requires 9 basic operations. Finally to complete step k of Algorithm 6, we must take one extra proximal map of g. Hence the number of basic operations needed to complete step k of Algorithm 6 is 9×(equation (2.44))+1, where we set $t = 1/\mu$.

Let us now compute the total cost across the outer iterations k. Theorem 2.5.6 shows that if we set $\varepsilon_k = \frac{1}{Lk^2}$ in each iteration k of Algorithm 5, then after N outer iterations we are guaranteed

$$\min_{j=0,\ldots,N-1} \left\|\mathcal{G}_{\frac{1}{\mu}}(x_j)\right\|^2 \leq \frac{4\mu\big(F(x_0) - F^* + 8\big)}{N}. \tag{2.45}$$

Thus we can find a point x satisfying

$$\left\|\mathcal{G}_{\frac{1}{\mu}}(x)\right\| \leq \varepsilon$$

after at most $\mathcal{N}(\varepsilon) := \left\lceil \frac{4\mu(F(x_0)-F^*+8)}{\varepsilon^2}\right\rceil$ outer-iterations and therefore after

$$\left\lceil\frac{4\mu(F(x_0) - F^* + 8)}{\varepsilon^2}\right\rceil \left(10 + 9\left\lceil\sqrt{\frac{\|\nabla c\|^2 L_h}{2\mu}} \cdot \log\left(\frac{8\|\nabla c\|^4 L^2(1+\mathcal{N}(\varepsilon))^4}{\beta^2}\right)\right\rceil\right) \tag{2.46}$$

basic operations in total. Thus the number of basic operations is on the order of

$$\boxed{\mathcal{O}\left(\frac{\sqrt{\|\nabla c\|^2\cdot L_h\cdot \mu}\cdot (F(x_0)-F^*)}{\varepsilon^2}\,\log\left(\frac{\|\nabla c\|^2 L^3\beta(F(x_0)-F^*)^2)}{\varepsilon^4}\right)\right)}\,. \tag{2.47}$$

Total cost based on approximate minimizers of the subproblems

Let us look at what goes wrong with applying Algorithm 3, with the proximal subproblems $\min_z F_t(z;x)$ approximately solved by a primal only method. To this end, notice that the objective function $F_t(\cdot;x)$ is a sum of the $\frac{1}{t}$-strongly convex and prox-friendly term $g+\frac{1}{2t}\|\cdot - x\|^2$ and the smooth convex function $z\mapsto h(c(x)+\nabla c(x)(z-x))$. The gradient of the smooth term is Lipschitz continuous with constant $\|\nabla c(x)\|_{\text{op}}^2 L_h$. Let us apply the fast gradient method (Algorithm 6) to the proximal subproblem directly. According to the estimate (2.42), Algorithm 6 will find an ε-approximate minimizer z of $F_t(\cdot;x)$ after at most

$$1+\sqrt{\frac{t\|\nabla c(x)\|_{\text{op}}^2 L_h}{2}}\cdot\log\left(\frac{\|\nabla c(x)\|_{\text{op}}^2 L_h\|x^*-z_0\|^2}{4\varepsilon}\right) \tag{2.48}$$

iterations, where x^* is the minimizer of $F_t(\cdot;x)$ and the scheme is initialized at z_0. The difficulty is that there appears to be no simple way to bound the distance $\|z_0-z^*\|$ for each proximal subproblem, unless we assume that dom g is bounded. We next show how we can correct for this difficulty by more carefully coupling the inexact prox-linear algorithm and the linearly convergent algorithm for solving the subproblem. In particular, in each outer iteration of the proposed scheme (Algorithm 7), one runs a linearly convergent subroutine $\mathcal{M}$ on the prox-linear subproblem for a fixed number of iterations; this fixed number of inner iterations depends explicitly on $\mathcal{M}$'s linear rate of convergence. The algorithmic idea behind this coupling originates in [70]. The most interesting consequence of this scheme is on so-called finite-sum problems, which we will discuss in Section 2.7. In this context, the algorithms that one runs on the proximal subproblems are stochastic. Consequently, we adapt our analysis to a stochastic setting as well, proving convergence rates on the expected norm of the prox-gradient $\|\mathcal{G}_t(x_k)\|$. When the proximal subproblems are approximately solved by deterministic methods, the convergence rates are all deterministic as well.

The following definition makes precise the types of algorithms that we will be able to accommodate as subroutines for the prox-linear subproblems.

Definition 2.6.2 (Linearly convergent subscheme)**.** A method $\mathcal{M}$ is *a linearly convergent subscheme* for the composite problem (2.9) if the following holds. For any points $x \in \mathbb{R}^d$, there exist constants $\gamma \geq 0$ and $\tau \in (0,1)$ so that when $\mathcal{M}$ is applied to $\min F_t(\cdot; x)$ with an arbitrary $z_0 \in \text{dom } g$ as an initial iterate, $\mathcal{M}$ generates a sequence $\{z_i\}_{i=1}^{\infty}$ satisfying

$$\mathbb{E}[F_t(z_i;x) - F_t(x^*;x)] \leq \gamma \, (1-\tau)^i \, \|z_0 - x^*\|^2 \qquad \text{for } i = 1, \ldots, \infty, \tag{2.49}$$

where x^* is the minimizer of $F_t(\cdot; x)$.

We apply a linearly convergent subscheme to proximal subproblems $\min F_t(\cdot; x_k)$, where x_k is generated in the previous iteration of an inexact prox-linear method. We then denote the resulting constants (γ, τ) in the guarantee (2.49) by (γ_k, τ_k).

The overall method we propose is Algorithm 7. It is important to note that in order to implement this method, one must know explicitly the constants (γ, τ) for the method $\mathcal{M}$ on each proximal subproblem.

Algorithm 7: Inexact prox-linear method: primal-only subsolves I

```
Initialize : A point $x_0 \in \text{dom } g$, real $t > 0$, a linearly convergent subscheme $\mathcal{M}$ for
             (2.9).
Step k: ($k \geq 1$)
Set $x_{k,0} := x_k$. Initialize $\mathcal{M}$ on the problem $\min_z F_t(z; x_k)$ at $x_{k,0}$, and run $\mathcal{M}$ for
        $T_k := \left\lceil \frac{1}{\tau_k} \log(4t\gamma_k) \right\rceil$    iterations,    (2.50)
thereby generating iterates $x_{k,1}, \ldots, x_{k,T_k}$.
Set $x_{k+1} = x_{k,T_k}$.
```

The following lemma shows that the proposed number of inner iterations (2.50) leads to significant progress in the prox-linear subproblems, compared with the initialization. Henceforth, we let $\mathbb{E}_{x_k}[\cdot]$ denote the expectation of a quantity conditioned on the iterate x_k.

Lemma 2.6.3. *The iterates x_k generated by Algorithm 7 satisfy*

$$\mathbb{E}_{x_k}[F_t(x_{k+1}; x_k) - F_t(x_k^*; x_k)] \leq \frac{1}{4t} \|x_k - x_k^*\|^2 . \tag{2.51}$$

Proof. In each iteration k, the linear convergence of algorithm $\mathcal{M}$ implies

$$\begin{aligned}\mathbb{E}_{x_k}[F_t(x_{k+1}; x_k) - F_t(x_k^*; x_k)] &\leq \gamma_k \left(1 - \tau_k\right)^{T_k} \|x_{k,0} - x_k^*\|^2 \\ &\leq \gamma_k e^{-\tau_k T_k} \|x_k - x_k^*\|^2 \\ &\leq \frac{1}{4t} \|x_k - x_k^*\|^2,\end{aligned}$$

as claimed. □

With this lemma at hand, we can establish convergence guarantees of the inexact method.

Theorem 2.6.4 (Convergence of Algorithm 7). *Supposing $t \leq \mu^{-1}$, the iterates x_k generated by Algorithm 7 satisfy*

$$\min_{j=0,\ldots,N-1} \mathbb{E}[\|\mathcal{G}_t(x_j)\|^2] \leq \frac{4t^{-1}\left(F(x_0) - \inf F\right)}{N}.$$

Proof. The proof follows the same outline as Theorem 2.5.2. Observe

$$\begin{aligned}\mathbb{E}_{x_k}[F(x_k) - F(x_{k+1})] &= \mathbb{E}_{x_k}[F_t(x_k; x_k) - F(x_{k+1})] \\ &\geq \mathbb{E}_{x_k}[F_t(x_k^*; x_k) - F(x_{k+1}) + \frac{1}{2t} \|x_k - x_k^*\|^2] \\ &\geq \mathbb{E}_{x_k}[F_t(x_k^*; x_k) - F_t(x_{k+1}; x_k)] + \frac{1}{2t} \|x_k - x_k^*\|^2 \\ &\geq -\frac{1}{4t} \|x_k - x_k^*\|^2 + \frac{1}{2t} \|x_k - x_k^*\|^2 \\ &\geq \frac{t}{4} \|\mathcal{G}_t(x_k)\|^2 ,\end{aligned}$$

where the second line follows from strong convexity of $F_t(\cdot; x_k)$, the third from Lemma 2.3.2, and the fourth from Lemma 2.6.3. Taking expectations of both sides, and using the tower rule, we deduce

$$\mathbb{E}[F(x_k) - F(x_{k+1})] \geq \frac{t}{4}\mathbb{E}[\|\mathcal{G}_t(x_k)\|^2].$$

Summing up both sides, we deduce

$$\begin{aligned}\min_{j=0,\ldots,N-1} \frac{t}{4}\, \mathbb{E}[\|\mathcal{G}_t(x_j)\|^2] &\leq \frac{t}{4N}\sum_{j=0}^{N-1} \mathbb{E}[\|\mathcal{G}_t(x_j)\|^2] \\ &\leq \frac{1}{N}\sum_{j=0}^{N-1} \mathbb{E}[F(x_j) - F(x_{j+1})] \\ &\leq \frac{F(x_0) - \inf F}{N},\end{aligned}$$

as claimed. □

It is clear from the proof that if the inner algorithm $\mathcal{M}$ satisfies (2.49) with the expectation $\mathbb{E}_{x_k}$ omitted, then Theorem 2.6.4 holds with $\mathbb{E}$ omitted as well and with $\inf F$ replaced by $F^* := \operatorname{liminf}_{k\to\infty} F(x_i)$. In particular, let us suppose that we set $t = \mu^{-1}$ and let $\mathcal{M}$ be the fast-gradient method (Algorithm 6) applied to the primal problem. Then in each iteration k, we can set $L_f = \|\nabla c(x_k)\|_{\mathrm{op}}^2 L_h$ and $\alpha = \mu$. Let us now determine γ_k and τ_k. Using the inequality $(1 + \sqrt{\frac{\alpha}{2L_f}})^{-1} \leq 1 - \sqrt{\frac{\alpha}{2L_f}}$ along with Theorem 2.6.1, we deduce we can set $\gamma_k = \frac{L_f}{4}$ and $\tau_k = \sqrt{\frac{\alpha}{2L_f}}$ for all indices k. Then each iteration of Algorithm 7 performs $T = \left\lceil \sqrt{\frac{2\|\nabla c(x_k)\|_{\mathrm{op}}^2 L_h}{\mu}} \log\left(\|\nabla c(x_k)\|_{\mathrm{op}}^2 L_h/\mu\right)\right\rceil$ iterations of the fast-gradient method, Algorithm 6. Recall that each iteration of Algorithm 6 requires 8 basic operations. Taking into account Theorem 7, we deduce that the overall scheme will produce a point x satisfying

$$\left\|\mathcal{G}_{\frac{1}{\mu}}(x)\right\| \leq \varepsilon$$

after at most

$$8\left\lceil\frac{4\mu\,(F(x_0) - F^*)}{\varepsilon^2}\right\rceil\left\lceil\sqrt{\frac{2\|\nabla c\|^2 L_h}{\mu}}\,\log\left(\frac{\|\nabla c\|^2 L_h}{\mu}\right)\right\rceil \tag{2.52}$$

basic operations. Thus the number of basic operations is on the order of

$$\boxed{\mathcal{O}\left(\frac{\sqrt{\|\nabla c\|^2 \cdot L_h \cdot \mu} \cdot (F(x_0) - F^*)}{\varepsilon^2}\ \log\left(\frac{\|\nabla c\|^2 L_h}{\mu}\right)\right)}. \tag{2.53}$$

Notice this estimate is better than (2.47), but only in terms of logarithmic dependence.

Before moving on, it is instructive to comment on the functional form of the linear convergence guarantee in (2.49). The right-hand-side depends on the initial squared distance $\|z_0 - x^*\|^2$. Convergence rates of numerous algorithms, on the other hand, are often stated with the right-hand-side instead depending on the initial functional error $F_t(z_0; x) - \inf_z F_t(z; x)$. In particular, this is the case for algorithms for finite sum problems discussed in Section 2.7, such as SVRG [55] and SAGA [34], and their accelerated extensions [1, 47, 69]. The following easy lemma shows how any such algorithm can be turned into a linearly convergent subscheme, in the sense of Definition 2.6.2, by taking a single extra prox-gradient step. We will use this observation in Section 2.7, when discussing finite-sum problems.

Lemma 2.6.5. *Consider an optimization problem having the convex additive composite form (2.38). Suppose $\mathcal{M}$ is an algorithm for $\min_z f^p(z)$ satisfying: there exist constants $\gamma \geq 0$ and $\tau \in (0, 1)$ so that on any input z_0, the method $\mathcal{M}$ generates a sequence $\{z_i\}_{i=1}^{\infty}$ satisfying*

$$\mathbb{E}[f^p(z_i) - f^p(z^*)] \leq \gamma\,(1-\tau)^i\,(f^p(z_0) - f^p(z^*)) \qquad \textit{for } i = 1, \ldots, \infty, \tag{2.54}$$

where z^* is a minimizer of f^p. Define an augmented method $\mathcal{M}^+$ as follows: given input z_0, initialize $\mathcal{M}$ at the point $\mathrm{prox}_{p/L_f}(z_0 - \frac{1}{L_f}\nabla f(z_0))$ and output the resulting points $\{z_i\}_{i=1}^{\infty}$. Then the iterates generates by $\mathcal{M}^+$ satisfy

$$\mathbb{E}[f^p(z_i) - f^p(z^*)] \leq \frac{\gamma L_f}{2}\,(1-\tau)^i\,\|z_0 - z^*\|^2 \qquad \textit{for } i = 1, \ldots, \infty,$$

Proof. Set $\hat{z} := \mathrm{prox}_{p/L_f}(z_0 - \frac{1}{L_f}\nabla f(z_0))$. Then convergence guarantees (2.54) of $\mathcal{M}$, with $\hat{z}$ in place of z_0, read

$$\mathbb{E}[f^p(z_i) - f^p(z^*)] \leq \gamma\,(1-\tau)^i\,(f^p(\hat{z}) - f^p(z^*)) \qquad \text{for } i = 1, \ldots, \infty.$$

Observe the inequality $f^p(\hat{z}) \leq f(z_0) + \langle \nabla f(z_0), \hat{z} - z_0 \rangle + p(\hat{z}) + \frac{L_f}{2}\|\hat{z} - z_0\|^2$. By definition, $\hat{z}$ is the minimizer of the function $z \mapsto f(z_0) + \langle \nabla f(z_0), z - z_0 \rangle + p(z) + \frac{L_f}{2}\|z - z_0\|^2$, and hence we deduce $f^p(\hat{z}) \leq f(z_0) + \langle \nabla f(z_0), z^* - z_0 \rangle + p(z^*) + \frac{L_f}{2}\|z^* - z_0\|^2 \leq f^p(z^*) + \frac{L_f}{2}\|z^* - z_0\|^2$, with the last inequality follows from convexity of f. The result follows. □

2.6.3 Total cost of the smoothing strategy

The final ingredient is to replace h by a smooth approximation and then minimize the resulting composite function by an inexact prox-linear method (Algorithms 5 or 7). Define the smoothed composite function

$$F^{\nu}(x) := g(x) + h_{\nu}(c(x)), \tag{2.55}$$

where h_ν is the Moreau envelope of h. Recall from Lemma 2.2.1 the three key properties of the Moreau envelope:

$$\operatorname{lip}(h_\nu) \leq L, \qquad \operatorname{lip}(\nabla h_\nu) \leq \frac{1}{\nu},$$

and

$$0 \leq h(z) - h_\nu(z) \leq \frac{L^2\nu}{2} \qquad \text{for all } z \in \mathbb{R}^m.$$

Indeed, these are the only properties of the smoothing we will use; therefore, in the analysis, any smoothing satisfying the analogous properties can be used instead of the Moreau envelope.

Let us next see how to choose the smoothing parameter $\nu > 0$ based on a target accuracy ε on the norm of the prox-gradient $\|\mathcal{G}_t(x)\|$. Naturally, we must establish a relationship between the step-sizes of the prox-linear steps on the original problem and its smooth approximation.

To distinguish between these two settings, we will use the notation

$$\begin{aligned} x^+ &= \operatorname*{argmin}_z \left\{h\big(c(x)+\nabla c(x)(z-x)\big)+g(z)+\tfrac{1}{2t}\|z-x\|^2\right\}, \\ \widehat{x} &= \operatorname*{argmin}_z \left\{h_\nu\big(c(x)+\nabla c(x)(z-x)\big)+g(z)+\tfrac{1}{2t}\|z-x\|^2\right\}, \\ \mathcal{G}_t(x) &= t^{-1}(x^+-x), \\ \mathcal{G}_t^\nu(x) &= t^{-1}(\widehat{x}-x). \end{aligned}$$

Thus $\mathcal{G}_t(x)$ is the prox-gradient on the target problem (2.9) as always, while $\mathcal{G}_t^\nu(x)$ is the prox-gradient on the smoothed problem (2.55). The following theorem will motivate our strategy for choosing the smoothing parameter ν.

Theorem 2.6.6 (Prox-gradient comparison)**.** *For any point x, the inequality holds:*

$$\|\mathcal{G}_t(x)\| \le \|\mathcal{G}_t^\nu(x)\| + \sqrt{\frac{L^2\nu}{2t}}.$$

Proof. Applying Lemma 2.2.1 and strong convexity of the proximal subproblems, we deduce

$$\begin{aligned} F_t(x^+;x) &\le F_t(\widehat{x};x) - \frac{1}{2t}\left\|\widehat{x}-x^+\right\|^2 \\ &\le \left(h_\nu\big(c(x)+\nabla c(x)(\widehat{x}-x)\big)+g(\widehat{x})+\frac{1}{2t}\|\widehat{x}-x\|^2\right)+\frac{L^2\nu}{2}-\frac{1}{2t}\left\|\widehat{x}-x^+\right\|^2 \\ &\le \left(h_\nu\big(c(x)+\nabla c(x)(x^+-x)\big)+g(x^+)+\frac{1}{2t}\left\|x^+-x\right\|^2\right)+\frac{L^2\nu}{2}-t^{-1}\left\|\widehat{x}-x^+\right\|^2 \\ &\le F_t(x^+;x)+\frac{L^2\nu}{2}-t^{-1}\left\|\widehat{x}-x^+\right\|^2. \end{aligned}$$

Canceling out like terms, we conclude $t^{-1}\|\widehat{x}-x^+\|^2 \le \frac{L^2\nu}{2}$. The triangle inequality then yields

$$t^{-1}\left\|x^+-x\right\| \le t^{-1}\|\widehat{x}-x\| + \sqrt{\frac{L^2\nu}{2t}},$$

as claimed.‘ □

Fix a target accuracy $\varepsilon > 0$. The strategy for choosing the smoothing parameter ν is now clear. Let us set $t = \frac{1}{\mu}$ and then ensure $\frac{\varepsilon}{2} = \sqrt{\frac{L^2\nu}{2t}}$ by setting $\nu := \frac{\varepsilon^2}{2L^3\beta}$. Then by Theorem 2.6.6, any point x satisfying $\|\mathcal{G}_{1/\mu}^\nu(x)\| \le \frac{\varepsilon}{2}$ would automatically satisfy the desired condition $\|\mathcal{G}_{1/\mu}(x)\| \le \varepsilon$. Thus we must only estimate the cost of obtaining such a point x. Following the discussion in Section 2.6.2, we can apply either of the Algorithms 5 or 7, along with the fast-gradient method (Algorithm 6) for the inner subsolves, to the problem $\min_x F^{\nu}(x) = g(x) + h_{\nu}(c(x))$. We note that for a concrete implementation, one needs the following formulas, complementing Lemma 2.5.3.

Lemma 2.6.7. *For any point x and real ν, $t > 0$, the following are true:*

$$prox_{th_\nu}(x) = (\tfrac{\nu}{t+\nu}) \cdot x + (\tfrac{t}{t+\nu}) \cdot prox_{(t+\nu)h}(x) \qquad and \qquad \nabla h_\nu(x) = \tfrac{1}{\nu}(x - prox_{\nu h}(x)).$$

Proof. The expression $\nabla h_\nu(x) = \frac{1}{\nu}(x - \mathrm{prox}_{\nu h}(x))$ was already recorded in Lemma 2.2.1. Observe the chain of equalities

$$\begin{aligned}\min_y \left\{ h_\nu(y) + \frac{1}{2t}\|y-x\|^2 \right\} &= \min_y \min_z \left\{ h(z) + \frac{1}{2\nu}\|z-y\|^2 + \frac{1}{2t}\|y-x\|^2 \right\} \qquad (2.56)\\ &= \min_z \left\{ h(z) + \frac{1}{2(t+\nu)}\|z-x\|^2 \right\},\end{aligned}$$

where the last equality follows by exchanging the two mins in (2.56). By the same token, taking the derivative with respect to y in (2.56), we conclude that the optimal pair (y,z) must satisfy the equality $0 = \nu^{-1}(y-z) + t^{-1}(y-x)$. Since the optimal y is precisely $\mathrm{prox}_{th_\nu}(x)$ and the optimal z is given by $\mathrm{prox}_{(t+\nu)h}(x)$, the result follows. □

Let us apply Algorithm 5 with the fast-gradient dual subsolves, as described in Section 2.6.2. Appealing to (2.46) with $L_h = \frac{1}{\nu} = \frac{2L^3\beta}{\varepsilon^2}$ and ε replaced by $\varepsilon/2$, we deduce that the scheme will find a point x satisfying $\|\mathcal{G}_{1/\mu}(x)\| \le \varepsilon$ after at most

$$\mathcal{N}(\varepsilon) \cdot \left(10 + 9 \left\lceil \frac{\|\nabla c\| L}{\varepsilon} \cdot \log\left(\frac{8\|\nabla c\|^4 L^2 (1+\mathcal{N}(\varepsilon))^4}{\beta^2} \right) \right\rceil \right)$$

basic operations, where $\mathcal{N}(\varepsilon) := \left\lceil \frac{16\mu\left(F(x_0) - \inf F + 8 + \frac{\varepsilon^2}{4\mu}\right)}{\varepsilon^2} \right\rceil$. Hence the total cost is on the order[4] of

$$\boxed{\mathcal{O}\left(\frac{L^2\beta\|\nabla c\| \cdot (F(x_0) - \inf F)}{\varepsilon^3} \log\left(\frac{\|\nabla c\|^2 L^3 \beta (F(x_0) - \inf F)^2}{\varepsilon^4} \right) \right)}\,. \qquad (2.57)$$

[4] Here, we use the assymptotic notation described in Remark 1 with $\omega = (\|\nabla c\|, L, \beta, F(x_0) - \inf F, 1/\varepsilon)$.

Similarly, let us apply Algorithm 7 with fast-gradient primal subsolves, as described in Section 2.6.2. Appealing to (2.52), we deduce that the scheme will find a point x satisfying $\|\mathcal{G}_{1/\mu}(x)\| \leq \varepsilon$ after at most

$$8\left\lceil \frac{16\mu\left(F(x_0) - \inf F + \frac{\varepsilon^2}{4\mu}\right)}{\varepsilon^2} \right\rceil \left\lceil \frac{2\|\nabla c\| L}{\varepsilon} \log\left(\frac{2\|\nabla c\|^2 L^2}{\varepsilon^2}\right) \right\rceil$$

basic operations. Thus the cost is on the order[4] of

$$\boxed{\mathcal{O}\left(\frac{L^2\beta\|\nabla c\| \cdot (F(x_0) - \inf F)}{\varepsilon^3} \log\left(\frac{\|\nabla c\| L}{\varepsilon}\right)\right)}. \tag{2.58}$$

Notice that the two estimates (2.57) and (2.58) are identical up to a logarithmic dependence on the problem data. To the best of our knowledge, these are the best-known efficiency estimates of any first-order method for the composite problem class (2.9).

2.7 Finite sum problems

In this section, we extend the results of the previous sections on so-called "finite sum problems", also often called "regularized empirical risk minimization". More precisely, throughout the section instead of minimizing a single composite function, we will be interested in minimizing an average of m composite functions:

$$\min_x \; F(x) := \frac{1}{m}\sum_{i=1}^{m} h_i(c_i(x)) + g(x). \tag{2.59}$$

In line with the previous sections, we make the following assumptions on the components of the problem:

1. g is a closed convex function;

2. $h_i \colon \mathbb{R} \to \mathbb{R}$ are convex, and L-Lipschitz continuous;

3. $c_i \colon \mathbb{R}^d \to \mathbb{R}$ are C^1-smooth with the gradient map ∇c_i that is β-Lipschitz continuous.

We also assume that we have available a real value, denoted $\|\nabla c\|$, satisfying

$$\|\nabla c\| \geq \sup_{x \in \mathrm{dom}\, g} \max_{i=1,\ldots,m} \|\nabla c_i(x)\|.$$

The main conceptual premise here is that m is large and should be treated as an explicit parameter of the problem. Moreover, notice the Lipschitz data is stated for the individual functional components of the problem. Such finite-sum problems are ubiquitous in machine learning and data science, where m is typically the (large) number of recorded measurements of the system. Notice that we have assumed that c_i maps to the real line. This is purely for notational convenience. Completely analogous results, as in this section, hold when c_i maps into a higher dimensional space.

Clearly, the finite-sum problem (2.59) is an instance of the composite problem class (2.9) under the identification

$$h(z_i, \ldots, z_m) := \frac{1}{m} \sum_{i=1}^{m} h_i(z_i) \qquad \text{and} \qquad c(x) := (c_1(x), \ldots, c_m(x)). \tag{2.60}$$

Therefore, given a target accuracy $\varepsilon > 0$, we again seek to find a point x with a small prox-gradient $\|\mathcal{G}_t(x)\| \leq \varepsilon$. In contrast to the previous sections, by a *basic operation* we will mean an individual gradient evaluation $\nabla c_i(x)$. In other words, we would like to find a point x with a small prox-gradient using as few individual gradient evaluations $\nabla c_i(x)$ as possible.

Let us next establish baseline efficiency estimates by simply using the inexact prox-linear schemes discussed in Sections 2.6.2 and 2.6.3. To this end, the following lemma derives Lipschitz constants of h and ∇c from the problem data L and β. The proof is elementary and we have placed it in Appendix A.1. Henceforth, we set $\mathrm{lip}\,(\nabla c) := \sup_{x \neq y} \frac{\|\nabla c(x) - \nabla c(y)\|_{\mathrm{op}}}{\|x-y\|}$.

Lemma 2.7.1 (Norm comparison)**.** *The inequalities hold:*

$$\mathrm{lip}\,(h) \leq L/\sqrt{m}, \qquad \mathrm{lip}\,(\nabla c) \leq \beta\sqrt{m}, \qquad \|\nabla c(x)\|_{op} \leq \sqrt{m}\left(\max_{i=1,\ldots,m} \|\nabla c_i(x)\|\right) \quad \forall x.$$

If in addition each h_i is C^1-smooth with L_h-Lipschitz derivative $t \mapsto h_i'(t)$, then the inequality, $\mathrm{lip}\,(\nabla h) \leq L_h/m$*, holds as well.*

Remark 2 (Notational substitution)**.** We will now apply the results of the previous sections to the finite sum problem (2.59) with h and c defined in (2.60). In order to correctly interpret results from the previous sections, according to Lemma 2.7.1, we must be mindful to replace L with $L/\sqrt{m}$, β with $\beta\sqrt{m}$, $\|\nabla c\|$ with $\sqrt{m}\|\nabla c\|$, and L_h with L_h/m. In particular, observe that we are justified in setting $\mu := L\beta$ without any ambiguity. Henceforth, we will be using this substitution routinely.

Baseline efficiency when h_i are smooth:

Let us first suppose that h_i are C^1-smooth with L_h-Lipschitz derivative and interpret the efficiency estimate (2.53). Notice that each gradient evaluation ∇c requires m individual gradient evaluations ∇c_i. Thus multiplying (2.53) by m and using Remark 2, the efficiency estimate (2.53) reads:

$$\boxed{\mathcal{O}\left(\frac{m\sqrt{\|\nabla c\|^2 \cdot L_h \cdot L \cdot \beta} \cdot (F(x_0) - \inf F)}{\varepsilon^2} \log\left(\frac{\|\nabla c\|^2 L_h}{L\beta}\right)\right)} \tag{2.61}$$

individual gradient evaluations of $\nabla c_i(\cdot)$.

Baseline efficiency when h_i are nonsmooth:

Now let us apply the smoothing technique described in Section 2.6.3. Multiplying the efficiency estimate (2.58) by m and using Remark 2 yields:

$$\boxed{\mathcal{O}\left(\frac{m \cdot L^2\beta\|\nabla c\| \cdot (F(x_0) - \inf F)}{\varepsilon^3} \log\left(\frac{\|\nabla c\| L}{\varepsilon}\right)\right)} \tag{2.62}$$

individual gradient evaluations of $\nabla c_i(\cdot)$.

The two displays (2.61) and (2.62) serve as baseline efficiency estimates, in terms of individual gradient evaluations ∇c_i, for obtaining a point x satisfying $\|\mathcal{G}_{1/\mu}(x)\| \leq \varepsilon$. We will now see that one can improve these guarantees in expectation. The strategy is perfectly in line with the theme of the paper. We will replace h by a smooth approximation if necessary, then apply an inexact prox-linear Algorithm 7, while approximately solving each subproblem by an "(accelerated) incremental method". Thus the only novelty here is a different scheme for approximately solving the proximal subproblems.

2.7.1 An interlude: incremental algorithms

There are a number of popular algorithms for finite-sum problems, including SAG [99], SAGA [34], SDCA [103], SVRG [55, 116], FINITO [33], and MISO [71]. All of these methods have similar linear rates of convergence, and differ only in storage requirements and in whether one needs to know explicitly the strong convexity constant. For the sake of concreteness, we will focus on SVRG following [116]. This scheme applies to finite-sum problems

$$\min_x \; f^p(x) := \frac{1}{m}\sum_{i=1}^m f_i(x) + p(x), \tag{2.63}$$

where p is a closed, α-strongly convex function ($\alpha > 0$) and each f_i is convex and C^1-smooth with ℓ-Lipschitz gradient ∇f_i. For notational convenience, define the condition number $\kappa := l/\alpha$. Observe that when each h_i is smooth, each proximal subproblem indeed has this form:

$$\min_z \; F_t(z;x) := \frac{1}{m}\sum_{i=1}^m h_i\Big(c_i(x) + \langle \nabla c_i(x), z - x\rangle\Big) + g(z) + \frac{1}{2t}\|z - x\|^2. \tag{2.64}$$

In Algorithm 8, we record the Prox-SVRG method of [116] for minimizing the function (2.63).

The following theorem from [116, Theorem 3.1] summarizes convergence guarantees of Prox-SVRG.

Theorem 2.7.2 (Convergence rate of Prox-SVRG)**.** *Algorithm 8, with the choices* $\eta = \frac{1}{10\ell}$ *and* $J = \lceil 100\kappa \rceil$, *will generate a sequence* $\{\widetilde{x}_s\}_{s\geq 1}$ *satisfying*

$$\mathbb{E}[f^p(\widetilde{x}_s) - f^p(x^*)] \leq 0.9^s (f^p(\widetilde{x}_0) - f^p(x^*)),$$

where x^* *is the minimizer of* f^p*. Moreover, each step* s *requires* $m + 2\lceil 100\kappa \rceil$ *individual gradient* ∇f_i *evaluations.*

Algorithm 8: The Prox-SVRG method [116]

```
Initialize : A point $\widetilde{x}_0 \in \mathbb{R}^d$, a real $\eta > 0$, a positive integer $J$.
Step s: $(s \geq 1)$
$\widetilde{x} = \widetilde{x}_{s-1}$;
$\widetilde{v} = \frac{1}{m}\sum_{i=1}^m \nabla f_i(\widetilde{x})$;
$x_0 = \widetilde{x}$
for $j = 1, 2, \ldots, J$ do
    pick $i_j \in \{1, \ldots, m\}$ uniformly at random
    $v_j = \widetilde{v} + (\nabla f_{i_j}(x_{j-1}) - \nabla f_{i_j}(\widetilde{x}))$
    $x_j = \mathrm{prox}_{\eta p}(x_{j-1} - \eta v_j)$
end
$\widetilde{x}_s = \frac{1}{J}\sum_{j=1}^J x_j$
```

Thus Prox-SVRG will generate a point x with $\mathbb{E}[f^p(x) - f^p(x^*)] \leq \varepsilon$ after at most

$$\mathcal{O}\left((m+\kappa)\ \log\left(\frac{f^p(\widetilde{x}_0) - f^p(x^*)}{\varepsilon}\right)\right) \tag{2.65}$$

individual gradient ∇f_i evaluations. It was a long-standing open question whether there is a method that improves the dependence of this estimate on the condition number κ. This question was answered positively by a number of algorithms, including Catalyst [69], accelerated SDCA [104], APPA [47], RPDG [56], and Katyusha [1]. For the sake of concreteness, we focus only on one of these methods, Katyusha [1]. This scheme follows the same epoch structure as SVRG, while incorporating iterate history. We summarize convergence guarantees of this method, established in [1, Theorem 3.1], in the following theorem.

Theorem 2.7.3 (Convergence rate of Katyusha)**.** *The Katyusha algorithm of [1] generates a sequence of iterates $\{\widetilde{x}_s\}_{s\geq 1}$ satisfying*

$$\frac{\mathbb{E}[f^p(\widetilde{x}_s) - f^p(x^*)]}{f^p(\widetilde{x}_0) - f^p(x^*)} \leq \begin{cases} 4\left(1+\sqrt{1/(6\kappa m)}\right)^{-2sm} & , \quad \text{if } \frac{m}{\kappa} \leq \frac{3}{8} \\ 3(1.5)^{-s} & , \quad \text{if } \frac{m}{\kappa} > \frac{3}{8} \end{cases}$$

where x^* is the minimizer of f^p. Moreover, each step s requires $3m$ individual gradient ∇f_i evaluations.[5]

To simplify the expression for the rate, using the inequality $(1+z)^m \geq 1+mz$ observe

$$\left(1+\sqrt{\tfrac{1}{6\kappa m}}\right)^{-2sm} \leq \left(1+\sqrt{\tfrac{2m}{3\kappa}}\right)^{-s}.$$

Using this estimate in Theorem 2.7.3 simplifies the linear rate to

$$\frac{\mathbb{E}[f^p(\widetilde{x}_s)-f^p(x^*)]}{f^p(\widetilde{x}_0)-f^p(x^*)} \leq 4\cdot\max\left\{\left(1+\sqrt{\tfrac{2m}{3\kappa}}\right)^{-s}, 1.5^{-s}\right\}.$$

Recall that each iteration of Katyusha requires $3m$ individual gradient ∇f_i evaluations. Thus the method will generate a point x with $\mathbb{E}[f^p(x)-f^p(x^*)] \leq \varepsilon$ after at most

$$\mathcal{O}\left(\left(m+\sqrt{m\kappa}\right)\ \log\left(\frac{f^p(\widetilde{x}_0)-f^p(x^*)}{\varepsilon}\right)\right)$$

individual gradient ∇f_i evaluations. Notice this efficiency estimate is significantly better than the guarantee (2.65) for Prox-SVRG only when $m \ll \kappa$. This setting is very meaningful in the context of smoothing. Indeed, since we will be applying accelerated incremental methods to proximal subproblems after a smoothing, the condition number κ of each subproblem can be huge.

Improved efficiency estimates when h_i are smooth:

Let us now suppose that each h_i is C^1-smooth with L_h-Lipschitz derivative h_i'. We seek to determine the efficiency of the inexact prox-linear method (Algorithm 7) that uses either Prox-SVRG or Katyusha as the linearly convergent subscheme $\mathcal{M}$. Let us therefore first look at the efficiency of Prox-SVRG and Katyusha on the prox-linear subproblem (2.64). Clearly we can set

$$\ell := L_h\cdot\left(\max_{i=1,\ldots,m}\|\nabla c_i(x)\|^2\right) \qquad \text{and} \qquad \alpha = t^{-1}.$$

[5] The constants 4 and 3 are hidden in the $\mathcal{O}$ notation in [1, Theorem 3.1]. They can be explicitly verified by following along the proof.

Notice that the convergence guarantees for Prox-SVRG and Katyusha are not in the standard form (2.49). Lemma 2.6.5, however, shows that they can be put into standard form by taking a single extra prox-gradient step in the very beginning of each scheme; we'll call these slightly modified schemes Prox-SVRG$^+$ and Katyusha$^+$. Taking into account Lemma 2.7.1, observe that the gradient of the function $z \mapsto h(c(x) + \nabla c(x)(z - x))$ is l-Lipschitz continuous. Thus according to Lemma 2.6.5, Prox-SVRG$^+$ and Katyusha$^+$ on input $\widetilde{z}_0$ satisfy

$$\mathbb{E}[F_t(\widetilde{z}_s; x) - F_t(z^*; x)] \leq \frac{\ell}{2} \cdot 0.9^s \|\widetilde{z}_0 - z^*\|^2 \qquad \text{for } s = 1, \ldots, \infty,$$

$$\mathbb{E}[F_t(\widetilde{z}_s; x) - F_t(z^*; x)] \leq \frac{4\ell}{2} \cdot \max\left\{ \left(1 + \sqrt{\tfrac{2m}{3\kappa}}\right)^{-s}, 1.5^{-s} \right\} \cdot \|\widetilde{z}_0 - z^*\|^2 \quad \text{for } s = 1, \ldots, \infty,$$

respectively, where z^* is the minimizer of $F_t(\cdot; x)$.

We are now ready to compute the total efficiency guarantees. Setting $t = 1/\mu$, Theorem 2.6.4 shows that Algorithm 7 will generate a point x with

$$\mathbb{E}\left[\|\mathcal{G}_{1/\mu}(x)\|^2\right] \leq \varepsilon^2$$

after at most $\left\lceil \frac{4\mu(F(x_0) - \inf F)}{\varepsilon^2} \right\rceil$ iterations. Each iteration k in turn requires at most

$$\left\lceil \frac{1}{\tau_k} \log(4t\gamma_k) \right\rceil \leq \left\lceil \frac{1}{0.1} \log\left(4 \cdot \frac{1}{\mu} \cdot \frac{L_h \cdot \|\nabla c\|^2}{2}\right) \right\rceil$$

iterations of Prox-SVRG$^+$ and at most

$$\left\lceil \frac{1}{\tau_k} \log(4t\gamma_k) \right\rceil \leq \left\lceil \max\left\{3, \left(1 + \sqrt{\tfrac{3L_h\|\nabla c\|^2}{2m\mu}}\right)\right\} \log\left(4 \cdot \frac{1}{\mu} \cdot \frac{4 \cdot L_h \cdot \|\nabla c\|^2}{2}\right) \right\rceil$$

iterations of Katyusha$^+$. Finally recall that each iteration s of Prox-SVRG$^+$ and Katyusha$^+$, respectively, requires $m + 2\left\lceil \frac{100 L_h \|\nabla c\|^2}{\mu} \right\rceil$ and $3m$ evaluations of individual ∇c_i. Hence the overall efficiency in terms of the number of individual gradient evaluations ∇c_i is on the order of

$$\boxed{\mathcal{O}\left(\frac{\left(\mu m + L_h \|\nabla c\|^2\right) \cdot (F(x_0) - \inf F)}{\varepsilon^2} \log\left(\frac{L_h \cdot \|\nabla c\|^2}{\mu} \right)\right)} \tag{2.66}$$

when using Prox-SVRG$^+$ and on the order of

$$\boxed{\mathcal{O}\left(\frac{\left(\mu m + \sqrt{\mu m L_h \|\nabla c\|^2}\right)\cdot (F(x_0) - \inf F)}{\varepsilon^2}\log\left(\frac{L_h \cdot \|\nabla c\|^2}{\mu}\right)\right)} \tag{2.67}$$

when using Katyusha$^+$. Notice that the estimate (2.67) is better than (2.66) precisely when $m \ll \frac{L_h\|\nabla c\|^2}{\mu}$.

Improved efficiency estimates when h_i are nonsmooth:

Finally, let us now no longer suppose that h_i are smooth in the finite-sum problem (2.59) and instead apply the smoothing technique. To this end, observe the equality

$$h_\nu(z) = \inf_y \left\{ \frac{1}{m}\sum_{i=1}^m h_i(y_i) + \frac{1}{2\nu}\|y - z\|^2 \right\} = \sum_{i=1}^m (h_i/m)_\nu(z_i).$$

Therefore the smoothed problem in (2.55) is also a finite-sum problem with

$$\min_x \ \frac{1}{m}\sum_{i=1}^m m\cdot (h_i/m)_\nu(c_i(x)) + g(x).$$

Thus we can can apply the convergence estimates we have just derived in the smooth setting with $h_i(t)$ replaced by $\phi_i(t) := m\cdot(h_i/m)_\nu(t)$. Observe that ϕ_i is L-Lipschitz by Lemma 2.2.1, while the derivative $\phi_i'(t) = m\cdot\nu^{-1}(t - \mathrm{prox}_{\frac{\nu}{m}h_i}(t))$ is Lipschitz with constant $L_h := \frac{m}{\nu}$. Thus according to the recipe following Theorem 2.6.6, given a target accuracy $\varepsilon > 0$ for the norm of the prox-gradient $\|\mathcal{G}_{\frac{1}{\mu}}(x)\|$, we should set

$$\nu := \frac{m\varepsilon^2}{2L^3\beta},$$

where we have used the substitutions dictated by Remark 2. Then Theorem 2.6.6 implies

$$\left\|\mathcal{G}_{1/\mu}(x)\right\| \le \left\|\mathcal{G}^{\nu}_{1/\mu}(x)\right\| + \frac{\varepsilon}{2} \qquad \text{for all } x,$$

where $\left\|\mathcal{G}^{\nu}_{1/\mu}(x)\right\|$ is the prox-gradient for the smoothed problem. Squaring and taking expectations on both sides, we can be sure $\mathbb{E}[\left\|\mathcal{G}_{1/\mu}(x)\right\|^2] \le \varepsilon^2$ if we find a point x satisfying

$\mathbb{E}\left[\left\|\mathcal{G}^{\nu}_{1/\mu}(x)\right\|^2\right] \leq \frac{\varepsilon^2}{4}$. Thus we must simply write the estimates (2.66) and (2.67) for the smoothed problem in terms of the original problem data. Thus to obtain a point x satisfying

$$\mathbb{E}[\left\|\mathcal{G}_{1/\mu}(x)\right\|^2] \leq \varepsilon^2,$$

it suffices to perform

$$\boxed{\mathcal{O}\left(\left(\frac{L\beta m}{\varepsilon^2} + \frac{L^2\beta\|\nabla c\|}{\varepsilon^3} \cdot \min\left\{\sqrt{m}, \frac{L\|\nabla c\|}{\varepsilon}\right\}\right)(F(x_0) - \inf F)\ \log\left(\frac{L\|\nabla c\|}{\varepsilon}\right)\right)} \tag{2.68}$$

individual gradient ∇c_i evaluations. The min in the estimate corresponds to choosing the better of the two, Prox-SVRG$^+$ and Katyusha$^+$, in each proximal subproblem in terms of their efficiency estimates. Notice that the $1/\varepsilon^3$ term in (2.68) scales only as $\sqrt{m}$. Therefore this estimate is an order of magnitude better than our baseline (2.62), which we were trying to improve. The caveat is of course that the estimate (2.68) is in expectation while (2.62) is deterministic.

2.8 An accelerated prox-linear algorithm

Most of the paper thus far has focused on the setting when the proximal subproblems (2) can only be approximately solved by first-order methods. On the other hand, in a variety of circumstances, it is reasonable to expect to solve the subproblems to a high accuracy by other means. For example, one may have available specialized methods for the proximal subproblems, or interior-point points methods may be available for moderate dimensions d and m, or it may be that case that computing an accurate estimate of $\nabla c(x)$ may already be the bottleneck (see e.g. Example 2.3.5). In this context, it is interesting to see if the basic prox-linear method can in some sense be "accelerated" by using inertial information. In this section, we do exactly that.

We propose an algorithm, motivated by the work of Ghadimi-Lan [48], that is adaptive to some natural constants measuring convexity of the composite function. This being said, the reader should keep in mind a downside the proposed scheme: our analysis (for the first time in the paper) requires the domain of g to be bounded. Henceforth, define

$$M := \sup_{x,y \in \mathrm{dom}\ g} \|x - y\|$$

and assume it to be finite.

To motivate the algorithm, let us first consider the additive composite setting (2.10) with $c(\cdot)$ in addition convex. Algorithms in the style of Nesterov's second accelerated method (see [80] or [109, Algorithm 1]) incorporate steps of the form $v_{k+1} = \mathrm{prox}_{tg}\left(v_k - t\nabla c(y_k)\right)$. That is, one moves from a point v_k in the direction of the negative gradient $-\nabla c(y_k)$ evaluated at a different point y_k, followed by a proximal operation. Equivalently, after completing a square one can write

$$v_{k+1} := \operatorname*{argmin}_z \left\{ c(y_k) + \langle \nabla c(y_k), z - v_k \rangle + \frac{1}{2t}\|z - v_k\|^2 + g(z) \right\}.$$

This is also the construction used by Ghadimi and Lan [48, Equation 2.37] for nonconvex additive composite problems. The algorithm we consider emulates this operation. There is a slight complication, however, in that the composite structure requires us to incorporate an additional scaling parameter α in the construction. We use the following notation:

$$\begin{aligned} F_\alpha(z; y, v) &:= g(z) + \frac{1}{\alpha} \cdot h\big(c(y) + \alpha \nabla c(y)(z - v)\big), \\ F_{t,\alpha}(z; y, v) &:= F_\alpha(z; y, v) + \frac{1}{2t}\|z - v\|^2, \\ S_{t,\alpha}(y, v) &:= \operatorname*{argmin}_z \ F_{t,\alpha}(z; y, v). \end{aligned}$$

Observe the equality $S_{t,1}(x, x) = S_t(x)$. In the additive composite setting, the mapping $S_{t,\alpha}(y, v)$ does not depend on α and the definition reduces to

$$S_{t,\alpha}(y, v) = \operatorname*{argmin}_z \left\{ c(y) + \langle \nabla c(y), z - v \rangle + \frac{1}{2t}\|z - v\|^2 + g(z) \right\} = \mathrm{prox}_{tg}\left(v - t\nabla c(y)\right).$$

The scheme we propose is summarized in Algorithm 9.

Remark 3 (Interpolation weights)**.** When L and β are unknown, one can instead equip Algorithm 9 with a backtracking line search. A formal description and the resulting convergence guarantees appear in Appendix A.2. We also note that instead of setting $a_k = \frac{2}{k+1}$,

```
Algorithm 9: Accelerated prox-linear method
  Initialize : Fix two points $x_0, v_0 \in \operatorname{dom} g$ and a real number $\tilde{\mu} > \mu$.
  Step k: $(k \geq 1)$ Compute
      $a_k = \frac{2}{k+1}$                                      (2.69)
      $y_k = a_k v_{k-1} + (1 - a_k) x_{k-1}$                    (2.70)
      $x_k = S_{1/\tilde{\mu}}(y_k)$                              (2.71)
      $v_k = S_{\frac{1}{\tilde{\mu} a_k}, a_k}(y_k, v_{k-1})$    (2.72)
```

one may use the interpolation weights used in FISTA [4]; namely, the sequence a_k may be chosen to satisfy the relation $\frac{1-a_k}{a_k^2} = \frac{1}{a_{k-1}^2}$, with similar convergence guarantees.

2.8.1 Convergence guarantees and convexity moduli

We will see momentarily that convergence guarantees of Algorithm 9 are adaptive to convexity (or lack thereof) of the composition $h \circ c$. To simplify notation, henceforth set

$$\Phi := h \circ c.$$

Weak convexity and convexity of the pair

It appears that there are two different convexity-like properties of the composite problem that govern convergence of Algorithm 9. The first is weak-convexity. Recall from Lemma 2.4.2 that Φ is ρ-weakly convex for some $\rho \in [0, \mu]$. Thus there is some $\rho \in [0, \mu]$ such that for any points $x, y \in \mathbb{R}^d$ and $a \in [0, 1]$, the approximate secant inequality holds:

$$\Phi(ax + (1-a)y) \leq a\Phi(x) + (1-a)\Phi(y) + \rho a(1-a)\|x - y\|^2.$$

Weak convexity is a property of the composite function $h \circ c$ and is not directly related to h nor c individually. In contrast, the algorithm we consider uses explicitly the composite structure. In particular, it seems that the extent to which the "linearization" $z \mapsto h(c(y) + \nabla c(y)(z-y))$ lower bounds $h(c(z))$ should also play a role.

Definition 2.8.1 (Convexity of the pair)**.** A real number $r > 0$ is called a *convexity constant of the pair* (h, c) *on a set* U if the inequality

$$h\big(c(y) + \nabla c(y)(z-y)\big) \leq h(c(z)) + \frac{r}{2}\|z-y\|^2 \qquad \text{holds for all } z, y \in U.$$

Inequalities (2.13) show that the pair (h, c) indeed has a convexity constant $r \in [0, \mu]$ on $\mathbb{R}^d$. The following relationship between convexity of the pair (h, c) and weak convexity of Φ will be useful.

Lemma 2.8.2 (Convexity of the pair implies weak convexity of the composition)**.** *If* r *is a convexity constant of* (h, c) *on a convex set* U*, then* Φ *is* r*-weakly convex on* U*.*

Proof. Suppose r is a convexity constant of (h, c) on U. Observe that the subdifferential of the convex function Φ and that of the linearization $h\big(c(y) + \nabla c(y)(\cdot - y)\big)$ coincide at $y = x$. Therefore a quick argument shows that for any $x, y \in U$ and $v \in \partial\Phi(y)$ we have

$$\Phi(x) \geq h(c(y) + \nabla c(y)(x-y)) - \frac{r}{2}\|x-y\|^2 \geq \Phi(y) + \langle v, x-y\rangle - \frac{r}{2}\|x-y\|^2.$$

The rest of the proof follows along the same lines as [30, Theorem 3.1]. We omit the details. □

Remark 4. The converse of the lemma is false. Consider for example setting $c(x) = (x, x^2)$ and $h(x, z) = x^2 - z$. Then the composition $h \circ c$ is identically zero and hence convex. On the other hand, one can easily check that the pair (h, c) has a nonzero convexity constant.

Convergence guarantees

Henceforth, let ρ be a weak convexity constant of $h \circ c$ on dom g and let r be a convexity constant of (h, c) on dom g. According to Lemma 2.8.2, we can always assume $0 \leq \rho \leq r \leq \mu$. We are now ready to state and prove convergence guarantees of Algorithm 9.

Theorem 2.8.3 (Convergence guarantees)**.** *Fix a real number $\tilde{\mu} > \mu$ and let x^* be any point satisfying $F(x^*) \leq F(x_k)$ for all iterates x_k generated by Algorithm 9. Then the efficiency estimate holds:*

$$\min_{j=1,\ldots,N} \left\|\mathcal{G}_{1/\tilde{\mu}}(y_j)\right\|^2 \leq \frac{24\tilde{\mu}^2}{\tilde{\mu}-\mu}\left(\frac{\tilde{\mu}\left\|x^*-v_0\right\|^2}{N(N+1)(2N+1)} + \frac{M^2(r+\frac{\rho}{2}(N+3))}{(N+1)(2N+1)}\right).$$

In the case $r = 0$, the inequality above holds with the second summand on the right-hand-side replaced by zero (even if $M = \infty$), and moreover the efficiency bound on function values holds:

$$F(x_N) - F(x^*) \leq \frac{2\tilde{\mu}\left\|x^*-v_0\right\|^2}{(N+1)^2}.$$

Succinctly, setting $\tilde{\mu} := 2\mu$, Theorem 2.8.3 guarantees the bound

$$\min_{j=1,\ldots,N} \left\|\mathcal{G}_{1/\tilde{\mu}}(y_j)\right\|^2 \leq \mathcal{O}\left(\frac{\mu^2\|x^*-v_0\|^2}{N^3}\right) + \frac{r}{\mu}\cdot\mathcal{O}\left(\frac{\mu^2 M^2}{N^2}\right) + \frac{\rho}{\mu}\cdot\mathcal{O}\left(\frac{\mu^2 M^2}{N}\right).$$

The fractions $0 \leq \frac{\rho}{\mu} \leq \frac{r}{\mu} \leq 1$ balance the three terms, corresponding to different levels of "convexity".

Our proof of Theorem 2.8.3 is based on two basic lemmas, as is common for accelerated methods [109].

Lemma 2.8.4 (Three-point comparison)**.** *Consider the point $z := S_{t,\alpha}(y,v)$ for some points $y, v \in \mathbb{R}^d$ and real numbers $t, \alpha > 0$. Then for all $w \in \mathbb{R}^d$ the inequality holds:*

$$F_\alpha(z;y,v) \leq F_\alpha(w;y,v) + \frac{1}{2t}\left(\|w-v\|^2 - \|w-z\|^2 - \|z-v\|^2\right).$$

Proof. This follows immediately by noting that the function $F_{t,\alpha}(\cdot;y,v)$ is strongly convex with constant $1/t$ and z is its minimizer by definition. □

Lemma 2.8.5 (Telescoping)**.** *Let a_k, y_k, x_k, and v_k be the iterates generated by Algorithm 9. Then for any point $x \in \mathbb{R}^d$ and any index k, the inequality holds:*

$$\begin{aligned} F(x_k) \le a_k F(x) {+} (1 - a_k) F(x_{k-1}) + \frac{\tilde{\mu} a_k^2}{2}(\|x - v_{k-1}\|^2 - \|x - v_k\|^2) \\ - \frac{\tilde{\mu} - \mu}{2}\|y_k - x_k\|^2 + \rho a_k \|x - x_{k-1}\|^2 + \frac{r a_k^2}{2}\|x - v_{k-1}\|^2. \end{aligned} \tag{2.73}$$

Proof. Notice that all the points x_k, y_k, and v_k lie in dom g. From inequality (2.13), we have

$$F(x_k) \le h\big(c(y_k) + \nabla c(y_k)(x_k - y_k)\big) + g(x_k) + \frac{\mu}{2}\left\|x_k - y_k\right\|^2. \tag{2.74}$$

Define the point $w_k := a_k v_k + (1 - a_k) x_{k-1}$. Applying Lemma 2.8.4 to $x_k = S_{1/\tilde{\mu},1}(y_k, y_k)$ with $w = w_k$ yields the inequality

$$\begin{aligned} h(c(y_k) + \nabla c(y_k)(x_k - y_k)) + g(x_k) \le h(c(y_k) + \nabla c(y_k)(w_k - y_k)) \\ + \frac{\tilde{\mu}}{2}(\|w_k - y_k\|^2 - \|w_k - x_k\|^2 - \|x_k - y_k\|^2) \\ + a_k g(v_k) + (1 - a_k) g(x_{k-1}). \end{aligned} \tag{2.75}$$

Note the equality $w_k - y_k = a_k(v_k - v_{k-1})$. Applying Lemma 2.8.4 again with $v_k = S_{\frac{1}{\tilde{\mu} a_k}, a_k}(y_k, v_{k-1})$ and $w = x$ yields

$$\begin{aligned} h\big(c(y_k) + a_k \nabla c(y_k)(v_k - v_{k-1})\big) + a_k g(v_k) \le h\big(c(y_k) + a_k \nabla c(y_k)(x - v_{k-1})\big) \\ + a_k g(x) + \frac{\tilde{\mu} a_k^2}{2}\left(\|x - v_{k-1}\|^2 - \|x - v_k\|^2 - \|v_k - v_{k-1}\|^2\right). \end{aligned} \tag{2.76}$$

Define the point $\hat{x} := a_k x + (1 - a_k) x_{k-1}$. Taking into account $a_k(x - v_{k-1}) = \hat{x} - y_k$, we conclude

$$\begin{aligned} h(c(y_k) + \nabla c(y_k)(\hat{x} - y_k)) &\le (h \circ c)(\hat{x}) + \frac{r}{2}\|\hat{x} - y_k\|^2 \\ &\le a_k h(c(x)) + (1 - a_k) h(c(x_{k-1})) \\ &\quad + \rho a_k (1 - a_k)\|x - x_{k-1}\|^2 + \frac{r a_k^2}{2}\|x - v_{k-1}\|^2. \end{aligned} \tag{2.77}$$

Thus combining inequalities (2.74), (2.75), (2.76), and (2.77), and upper bounding $1 - a_k \le 1$ and $-\|w_k - x_k\|^2 \le 0$, we obtain

$$\begin{aligned} F(x_k) \le a_k F(x) + (1 - a_k) F(x_{k-1}) + \frac{\tilde{\mu} a_k^2}{2}(\|x - v_{k-1}\|^2 - \|x - v_k\|^2) \\ - \frac{\tilde{\mu} - \mu}{2}\|y_k - x_k\|^2 + \rho a_k \|x - x_{k-1}\|^2 + \frac{r a_k^2}{2}\|x - v_{k-1}\|^2. \end{aligned}$$

The proof is complete. □

The proof of Theorem 2.8.3 now quickly follows.

Proof of Theorem 2.8.3. Set $x = x^*$ in inequality (2.73). Rewriting (2.73) by subtracting $F(x^*)$ from both sides, we obtain

$$\frac{F(x_k) - F(x^*)}{a_k^2} + \frac{\tilde{\mu}}{2}\left\|x^* - v_k\right\|^2 \leq \frac{1 - a_k}{a_k^2}\big(F(x_{k-1}) - F(x^*)\big) + \frac{\tilde{\mu}}{2}\left\|x^* - v_{k-1}\right\|^2 + \frac{\rho M^2}{a_k} + \frac{rM^2}{2} - \frac{\tilde{\mu} - \mu}{2a_k^2}\left\|x_k - y_k\right\|^2. \tag{2.78}$$

Using the inequality $\frac{1-a_k}{a_k^2} \leq \frac{1}{a_{k-1}^2}$ and recursively applying the inequality above N times, we get

$$\frac{F(x_N) - F(x^*)}{a_N^2} + \frac{\tilde{\mu}}{2}\|x^* - v_N\|^2 \leq \frac{1 - a_1}{a_1^2}\big(F(x_0) - F(x^*)\big) + \frac{\tilde{\mu}}{2}\left\|x^* - v_0\right\|^2 + \rho M^2\left(\sum_{j=1}^{N}\frac{1}{a_j}\right) + \frac{NrM^2}{2} - \frac{\tilde{\mu} - \mu}{2}\sum_{j=1}^{N}\frac{\left\|x_j - y_j\right\|^2}{a_j^2}. \tag{2.79}$$

Noting $F(x_N) - F(x^*) > 0$ and $a_1 = 1$, we obtain

$$\frac{\tilde{\mu} - \mu}{2}\sum_{j=1}^{N}\frac{\left\|x_j - y_j\right\|^2}{a_j^2} \leq \frac{\tilde{\mu}}{2}\left\|x^* - v_0\right\|^2 + \rho M^2\left(\sum_{j=1}^{N}\frac{1}{a_j}\right) + \frac{NrM^2}{2} \tag{2.80}$$

and hence

$$\frac{\tilde{\mu} - \mu}{2}\left(\sum_{j=1}^{N}\frac{1}{a_j^2}\right)\min_{j=1,\ldots,N}\left\|x_j - y_j\right\|^2 \leq \frac{\tilde{\mu}}{2}\left\|x^* - v_0\right\|^2 + \rho M^2\left(\sum_{j=1}^{N}\frac{1}{a_j}\right) + \frac{NrM^2}{2}.$$

Using the definition $a_k = \frac{2}{k+1}$, we conclude

$$\sum_{j=1}^{N}\frac{1}{a_j^2} = \frac{1}{4}\sum_{j=1}^{N}(j+1)^2 \geq \frac{1}{4}\sum_{j=1}^{N}j^2 = \frac{N(N+1)(2N+1)}{24}$$

and

$$\sum_{j=1}^{N}\frac{1}{a_j} = \sum_{j=1}^{N}\frac{j+1}{2} = \frac{N(N+3)}{4}.$$

With these bounds, we finally deduce

$$\min_{j=1,\ldots N} \|x_j - y_j\|^2 \leq \frac{24}{\tilde{\mu} - \mu} \left(\frac{\tilde{\mu} \|x^* - v_0\|^2}{N(N+1)(2N+1)} + \frac{M^2(r + \frac{\rho}{2}(N+3)}{(N+1)(2N+1)} \right),$$

thereby establishing the first claimed efficiency estimate in Theorem 2.8.3.

Finally suppose $r = 0$, and hence we can assume $\rho = 0$ by Lemma 2.8.2. Inequality (2.79) then becomes

$$\frac{F(x_N) - F(x^*)}{a_N^2} + \frac{\tilde{\mu}}{2} \|x^* - v_N\|^2 \leq \frac{\tilde{\mu}}{2} \|x^* - v_0\|^2 - \frac{\tilde{\mu} - \mu}{2} \sum_{j=1}^{N} \frac{\|x_j - y_j\|^2}{a_j^2}.$$

Dropping terms, we deduce $\frac{F(x_N)-F(x^*)}{a_N^2} \leq \frac{\tilde{\mu}}{2} \|x^* - v_0\|^2$, and the claimed efficiency estimate follows. □

2.8.2 Inexact computation

Completely analogously, we can consider an inexact accelerated prox-linear method based on approximately solving the duals of the prox-linear subproblems (Algorithm 10).

Theorem 2.8.6 (Convergence of inexact accelerated prox-linear method: near- stationarity)**.**

Fix a real number $\tilde{\mu} \geq \mu$ and let x^* be any point satisfying $F(x^*) \leq F(x_k)$ for iterates x_k generated by Algorithm 10. Then for any $N \geq 1$, the iterates x_k satisfy the inequality:

$$\min_{i=1,\ldots,N} \|\mathcal{G}_{1/\tilde{\mu}}(y_j)\|^2 \leq \frac{48\tilde{\mu}^2}{\tilde{\mu} - \mu} \left(\frac{\|x^* - v_0\|^2 + 4L \sum_{j=1}^{N} \frac{2\varepsilon_j + \delta_j}{a_j^2}}{N(N+1)(2N+1)} + \frac{M^2(r + \frac{\rho}{2}(N+3))}{(N+1)(2N+1)} \right).$$

Moreover, in the case $r = 0$, the inequality above holds with the second summand on the right-hand-side replaced by zero (even if $M = \infty$) and the following complexity bound on function values holds:

$$F(x_N) - F(x^*) \leq \frac{2\tilde{\mu}\|v_0 - x^*\|^2 + 8L \sum_{j=1}^{N} \frac{\varepsilon_j + \delta_j}{a_j^2}}{(N+1)^2}.$$

```
Algorithm 10: Inexact accelerated prox-linear method: near-stationarity
  Initialize : Fix two points $x_0, v_0 \in \operatorname{dom} g$ and a real number $\tilde{\mu} > \mu$.
  Step k: $(k \geq 1)$ Compute
      $a_k = \frac{2}{k+1}$
      $y_k = a_k v_{k-1} + (1 - a_k) x_{k-1}$
  - Find $(x_k, \zeta_k)$ such that $\|\zeta_k\| \leq \varepsilon_k$ and $x_k$ is the minimizer of the function
      $z \mapsto g(z) + h\Big(\zeta_k + c(y_k) + \nabla c(y_k)(z - y_k)\Big) + \frac{\tilde{\mu}}{2}\|z - y_k\|^2.$   (2.81)
  - Find $(v_k, \xi_k)$ such that $\|\xi_k\| \leq \delta_k$ and $v_k$ is the minimizer of the function
      $v \mapsto g(v) + \frac{1}{a_k} h\Big(\xi_k + c(y_k) + a_k \nabla c(y_k)(v - v_{k-1})\Big) + \frac{\tilde{\mu} a_k}{2}\|v - v_{k-1}\|^2.$   (2.82)
```

The proof appears in Appendix A.1. Thus to preserve the rate in N of the exact accelerated prox-linear method in Theorem 2.8.3, it suffices to require the sequences $\frac{\varepsilon_j}{a_j^2}, \frac{\delta_j}{a_j^2}$ to be summable. Hence we can set $\varepsilon_j, \delta_j \sim \frac{1}{j^{3+q}}$ for some $q > 0$.

Similarly, we can consider an inexact version of the accelerated prox-linear method based on approximately solving the primal problems in function value. The scheme is recorded in Algorithm 11.

Theorem 2.8.7 presents convergence guarantees of Algorithm 11. The statement of Theorem 2.8.7 is much more cumbersome than the analogous Theorem 2.8.6. The only take-away message for the reader is that to preserve the rate of the exact accelerated prox-linear method in Theorem 2.8.3 in terms of N, it sufficies for the sequences $\{\sqrt{i\delta_i}\}$, $\{i\delta_i\}$, and $\{i^2\varepsilon_i\}$ to be summable. Thus it suffices to take $\varepsilon_i, \delta_i \sim \frac{1}{i^{3+q}}$ for some $q > 0$.

The proof of Theorem 2.8.7 appears in Appendix A.1. Analysis of inexact accelerated

```
Algorithm 11: Accelerated prox-linear method: near-optimality
  Initialize : Fix two points $x_0, v_0 \in \operatorname{dom} g$, a real number $\tilde{\mu} > L\beta$, and two sequences
               $\varepsilon_i, \delta_i \geq 0$ for $i = 1, 2, \ldots, \infty$.
  Step k: ($k \geq 1$) Compute
        $a_k = \frac{2}{k+1}$                                                          (2.83)
        $y_k = a_k v_{k-1} + (1 - a_k) x_{k-1}$                                        (2.84)
        Set $x_k$ to be a $\varepsilon_k$-approximate minimizer of $F_{1/\tilde{\mu}}(\cdot; y_k)$                (2.85)
        Set $v_k$ to be a $\delta_k$-approximate minimizer of $F_{\frac{1}{\tilde{\mu} a_k}, a_k}(\cdot; y_k, v_{k-1})$      (2.86)
```

methods of this type for additive convex composite problems has appeared in a variety of papers, including [69, 100, 110]. In particular, our proof shares many features with that of [100], relying on approximate subdifferentials and the recurrence relation [100, Lemma 1].

Theorem 2.8.7 (Convergence of the accelerated prox-linear algorithm: near-optimality)**.** *Fix a real number $\tilde{\mu} > \mu$, and let x^* be any point satisfying $F(x^*) \leq F(x_k)$ for iterates x_k generated by Algorithm 11. Then the iterates x_k satisfy the inequality:*

$$\min_{i=1,\ldots,N} \|\mathcal{G}_{1/\tilde{\mu}}(y_i)\|^2 \leq \frac{96\tilde{\mu}^2}{\tilde{\mu} - \mu}\left(\frac{\tilde{\mu}\|x^* - v_0\|^2}{2N(N+1)(2N+1)} + + \frac{M^2(r + \frac{\rho}{2}(N+3))}{2(N+1)(2N+1)} \right.$$
$$\left. + \frac{\sum_{i=1}^{N}(\frac{\delta_i a_i + 3\varepsilon_i}{a_i^2}) + A_N\sqrt{2\tilde{\mu}}\sum_{i=1}^{N}\sqrt{\frac{\delta_i}{a_i}}}{N(N+1)(2N+1)} \right)$$

with

$$A_N := \sqrt{\tfrac{2}{\tilde{\mu}}}\sum_{i=1}^{N}\sqrt{\frac{\delta_i}{a_i}} + \left(\|x^* - v_0\|^2 + \tfrac{M^2 N(r + \frac{\rho}{2}(N+3))}{\tilde{\mu}} + \tfrac{2}{\tilde{\mu}}\sum_{i=1}^{N}\tfrac{\delta_i a_i + 2\varepsilon_i}{a_i^2} + \tfrac{2}{\tilde{\mu}}\left(\sum_{i=1}^{N}\sqrt{\frac{\delta_i}{a_i}}\right)^2 \right)^{1/2}.$$

Moreover, in the case $r = 0$, the inequality above holds with the second summand on the right-hand-side replaced by zero (even if $M = \infty$), and the following complexity bound on *function values holds:*

$$F(x_N) - F(x^*) \leq \frac{2\tilde{\mu}\left\|x^* - v_0\right\|^2 + 4\sum_{i=1}^{N}\frac{\delta_i a_i + 2\varepsilon_i}{a_i^2} + 4A_N\sqrt{2\tilde{\mu}}\sum_{i=1}^{N}\sqrt{\frac{\delta_i}{a_i}}}{(N+1)^2}.$$

Note that with the choices ε_i, $\delta_i \sim \frac{1}{i^{3+q}}$, the quantity A_N remains bounded. Consequently, in the setting $r = 0$, the functional error $F(x_N) - F(x^*)$ is on the order of $\mathcal{O}(1/N^2)$.

Chapter 3

4WD-CATALYST ACCELERATION FOR GRADIENT-BASED NON-CONVEX OPTIMIZATION

Joint work with H. Lin, D. Drusvyatskiy, J. Mairal, and Z. Harchaoui [88]

Abstract. We introduce a generic scheme to solve nonconvex optimization problems using gradient-based algorithms originally designed for minimizing convex functions. When the objective is convex, the proposed approach enjoys the same properties as the Catalyst approach of Lin et al. [69]. When the objective is nonconvex, it achieves the best known convergence rate to stationary points for first-order methods. Specifically, the proposed algorithm does not require knowledge about the convexity of the objective; yet, it obtains an overall worst-case efficiency of $\widetilde{O}(\varepsilon^{-2})$ and, if the function is convex, the complexity reduces to the near-optimal rate $\widetilde{O}(\varepsilon^{-2/3})$. We conclude the paper by showing promising experimental results obtained by applying the proposed approach to SVRG and SAGA for sparse matrix factorization and for learning neural networks.

3.1 Introduction

We consider optimization problems of the form

$$\min_{x \in \mathbb{R}^p} \left\{ f(x) := f_0(x) + \psi(x) \right\} , \quad \text{where } f_0(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \ . \tag{3.1}$$

Here, each function $f_i \colon \mathbb{R}^p \to \mathbb{R}$ is smooth, the regularization $\psi \colon \mathbb{R}^p \to \overline{\mathbb{R}}$ may be nonsmooth, and $\overline{\mathbb{R}} := \mathbb{R} \cup \{\infty\}$. By considering extended-real-valued functions, this composite setting also encompasses constrained minimization by letting ψ be the indicator function of the constraints on x. Minimization of regularized empirical risk objectives of form (3.1) is central in machine learning. Whereas a significant amount of work has been devoted to this composite setting for convex problems, leading in particular to fast incremental algorithms [see, *e.g.*, 34, 56, 71, 99, 114, 116], the question of minimizing efficiently (3.1) when the functions f_i and ψ may be nonconvex is still largely open today.

Yet, nonconvex problems in machine learning are of high interest. For instance, the variable x may represent the parameters of a neural network, where each term $f_i(x)$ measures the fit between x and a data point indexed by i, or (3.1) may correspond to a nonconvex matrix factorization problem (see Section 3.6). Besides, even when the data-fitting functions f_i are convex, it is also typical to consider nonconvex regularization functions ψ, for example for feature selection in signal processing [52]. In this work, we address two questions from nonconvex optimization:

1. How to apply a method for convex optimization to a nonconvex problem?

2. How to design an algorithm which does not need to know whether the objective function is convex while obtaining the optimal convergence guarantee if the function is convex?

Several pioneering works attempted to transfer ideas from the convex world to the nonconvex one, see, *e.g.*, [48, 49]. Our paper has a similar goal and studies the extension of Nesterov's acceleration for convex problems [79] to nonconvex composite ones. Unfortunately, the concept of acceleration for nonconvex problems is unclear from a worst-case complexity point of view: gradient descent requires $O(\varepsilon^{-2})$ iterations to guarantee a gradient norm smaller than ε [20, 21]. Under a stronger assumption that the objective function is C^2-smooth, state-of-the-art methods [*e.g.*, 18] achieve a marginal gain with complexity $O(\varepsilon^{-7/4}\log(1/\varepsilon))$, and do not appear to generalize to composite or finite-sum settings. For this reason, our work fits within a broader stream of recent research on methods that *do not perform worse than gradient descent in the nonconvex case* (in terms of worst-case complexity), while *automatically accelerating for minimizing convex functions.* The hope when applying such methods to nonconvex problems is to see acceleration in practice, by heuristically exploiting convexity that is "hidden" in the objective (for instance, local convexity near the optimum, or convexity along the trajectory of iterates).

The main contribution of this paper is a *generic* meta-algorithm, dubbed 4WD-Catalyst-Automatic, which is able to use a *gradient-based* optimization method $\mathcal{M}$, originally designed for convex problems, and turn it into an accelerated scheme that also applies to nonconvex objective functions. The proposed 4WD-Catalyst-Automatic can be seen as a **4**-**W**heel-**D**rive extension of Catalyst [69] to all optimization "terrains" (convex and nonconvex), while Catalyst was originally proposed for convex optimization. Specifically, without knowing whether the objective function is convex or not, our algorithm may take a method $\mathcal{M}$ designed for convex optimization problems with the same structure as (3.1), *e.g.*, SAGA [34], SVRG [116], and apply $\mathcal{M}$ to a sequence of sub-problems such that it asymptotically provides a stationary point of the nonconvex objective. Overall, the number of iterations of $\mathcal{M}$ to obtain a gradient norm smaller than ε is $\widetilde{O}(\varepsilon^{-2})$ in the worst case, while automatically reducing to $\widetilde{O}(\varepsilon^{-2/3})$ if the function is convex.[1]

Related work. Inspired by Nesterov's acceleration method for convex optimization [81], the first accelerated method performing universally well for nonconvex and convex problems was introduced in [48]. Specifically, the work [48] addresses composite problems such as (3.1) with $n = 1$, and, provided the iterates are bounded, it performs no worse than gradient descent on nonconvex instances with complexity $O(\varepsilon^{-2})$ on the gradient norm. When the problem is convex, it accelerates with complexity $O(\varepsilon^{-2/3})$. Extensions to accelerated Gauss-Newton type methods were also recently developed in [39]. In a follow-up work [49], a new scheme is proposed, which monotonically interlaces proximal gradient descent steps and Nesterov's extrapolation; thereby achieving similar guarantees as [48] but without the need to assume the iterates to be bounded. Extensions when the gradient of ψ is only Hölder continuous can also be devised.

In [67], a similar strategy is proposed, focusing instead on convergence guarantees under the so-called Kurdyka-Łojasiewicz inequality—a property corresponding to polynomial-like

[1] In this section, the notation $\widetilde{O}$ only displays the polynomial dependency with respect to ε for the clarity of exposition.

growth of the function, as shown by [7]. Our scheme is in the same spirit as these previous papers, since it monotonically interlaces proximal-point steps (instead of proximal-gradient as in [49]) and extrapolation/acceleration steps. A fundamental difference is that our method is generic and accommodates inexact computations, since we allow the subproblems to be approximately solved by any method we wish to accelerate.

By considering C^2-smooth nonconvex objective functions f with Lipschitz continuous gradient ∇f and Hessian $\nabla^2 f$, Carmon et al. [18] propose an algorithm with complexity $O(\varepsilon^{-7/4}\log(1/\varepsilon))$, based on iteratively solving convex subproblems closely related to the original problem. It is not clear if the complexity of their algorithm improves in the convex setting. Note also that the algorithm proposed in [18] is inherently for C^2-smooth minimization and requires exact gradient evaluations. This implies that the scheme does not allow incorporating nonsmooth regularizers and can not exploit finite sum structure.

Finally, a stochastic method related to SVRG [55] for minimizing large sums while automatically adapting to the weak convexity constant of the objective function is proposed in [2]. When the weak convexity constant is small (*i.e.*, the function is nearly convex), the proposed method enjoys an improved efficiency estimate. This algorithm, however, does not automatically accelerate for convex problems, in the sense that the overall rate is slower than $O(\varepsilon^{-2/3})$ in terms of target accuracy ε on the gradient norm.

Organization of the paper. Section 3.2 presents mathematical tools for non-convex and non-smooth analysis, which are used throughout the paper. In Sections 3.3 and 3.4, we introduce the main algorithm and important extensions, respectively. Finally, we present experimental results on matrix factorization and training of neural networks in Section 3.6.

3.2 Tools for nonconvex and nonsmooth optimization

Convergence results for nonsmooth optimization typically rely on the concept of subdifferential, which does not admit a unique definition in a nonconvex context [9]. In this paper, we circumvent this issue by focusing on a broad class of nonconvex functions known as weakly convex or lower C^2 functions, for which all these constructions coincide. Weakly convex functions cover most of the interesting cases of interest in machine learning and resemble convex functions in many aspects. In this section, we formally introduce them and discuss their subdifferential properties.

Definition 3.2.1 (Weak convexity)**.** A function $f\colon \mathbb{R}^p \to \overline{\mathbb{R}}$ is $\rho-$*weakly convex* if for any points $x, y \in \mathbb{R}^p$ and $\lambda \in [0,1]$, the approximate secant inequality holds:

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y) + \rho\lambda(1-\lambda)\left\|x-y\right\|^2.$$

Notice that ρ-weak convexity with $\rho = 0$ is exactly the definition of a convex function. An elementary algebraic manipulation shows that f is ρ-weakly convex if and only if the function $x \mapsto f(x) + \frac{\rho}{2}\left\|x\right\|^2$ is convex. In particular, a C^1-smooth function f is ρ-weakly convex if the gradient ∇f is ρ-Lipschitz, while a C^2-smooth function f is ρ-weakly convex if and only if $\nabla^2 f(x) \succeq -\rho I$ for all x. This closely resembles an equivalent condition for C^2-smooth and μ-strongly convex functions, namely $\nabla^2 f(x) \succeq \mu I$ with $\mu > 0$.

Useful characterizations of ρ-weakly convex functions rely on differential properties. Since the functions we consider in the paper are nonsmooth, we use a generalized derivative construction. We mostly follow the standard monograph on the subject by Rockafellar and Wets [98].

Definition 3.2.2 (Subdifferential)**.** Consider a function $f\colon \mathbb{R}^p \to \overline{\mathbb{R}}$ and a point x with $f(x)$ finite. The *subdifferential* of f at x is the set

$$\partial f(x) := \{v \in \mathbb{R}^p : f(y) \geq f(x) + v^T(y-x) + o(\|y-x\|) \quad \forall y \in \mathbb{R}^p\}.$$

Thus, a vector v lies in $\partial f(x)$ whenever the linear function $y \mapsto f(x) + v^T(y-x)$ is a lower-model of f, up to first-order around x. In particular, the subdifferential $\partial f(x)$ of a differentiable function f is the singleton $\{\nabla f(x)\}$, while for a convex function f it coincides with the subdifferential in the sense of convex analysis [see 98, Exercise 8.8]. It is useful to keep in mind that the sum rule, $\partial(f+g)(x) = \partial f(x) + \nabla g(x)$, holds for any differentiable function g.

We are interested in deriving complexity bounds on the number of iterations required by a method $\mathcal{M}$ to guarantee

$$\operatorname{dist}\big(0, \partial f(x)\big) \leq \varepsilon \ .$$

Recall when $\varepsilon = 0$, we are at a stationary point and satisfy first-order optimality conditions. In our convergence analysis, we will also use the following differential characterization of ρ-weakly convex functions, which generalize classical properties of convex functions. A proof follows directly from Theorem 12.17 of [98] by taking into account that f is ρ-weakly convex if and only if $f + \frac{\rho}{2}\|\cdot\|^2$ is convex.

Theorem 3.2.3 (Differential characterization of ρ-weakly convex functions)**.** *For any lower-semicontinuous function $f \colon \mathbb{R}^p \to \overline{\mathbb{R}}$, the following properties are equivalent:*

1. *f is ρ-weakly convex.*

2. **(subgradient inequality).** *For all x, y in $\mathbb{R}^p$ and v in $\partial f(x)$, we have*

$$f(y) \geq f(x) + v^T(y - x) - \frac{\rho}{2}\|y - x\|^2 .$$

3. **(hypo-monotonicity).** *For all x, y in $\mathbb{R}^p$, v in $\partial f(x)$, and w in $\partial f(y)$,*

$$(v - w)^T(x - y) \geq -\rho\|x - y\|^2.$$

Weakly convex functions have appeared in a wide variety of contexts, and under different names. Some notable examples are globally lower-C^2 [97], prox-regular [91], proximally smooth functions [25], and those functions whose epigraph has positive reach [45].

3.3 The 4WD-Catalyst algorithm for non-convex optimization

We now present a generic scheme (Algorithm 12) for applying a convex optimization method to minimize

$$\min_{x \in \mathbb{R}^p} \ f(x), \tag{3.2}$$

where f is only ρ-weakly convex. Our goal is to develop a unified framework that automatically accelerates in convex settings. Consequently, the scheme must be agnostic to the constant ρ.

3.3.1 4WD-Catalyst : a meta algorithm

At the center of our meta algorithm (Algorithm 12) are two sequences of subproblems obtained by adding simple quadratics to f. The proposed approach extends the Catalyst acceleration of [69] and comes with a simplified convergence analysis. We next describe in detail each step of the scheme.

Two-step subproblems. The proposed acceleration scheme builds two main sequences of iterates $(\bar{x}_k)_k$ and $(\tilde{x}_k)_k$, obtained from approximately solving two subproblems. These subproblems are simple quadratic perturbations of the original problem f having the form:

$$\min_x \left\{ f_\kappa(x; y) := f(x) + \frac{\kappa}{2} \|x - y\|^2 \right\}.$$

Here, κ is a regularization parameter and y is called the *prox-center*. By adding the quadratic, we make the problem more "convex": when f is non convex, with a large enough κ, the subproblem will be convex; when f is convex, we improve the conditioning of the problem.

At the k-th iteration, given a previous iterate x_{k-1} and the extrapolation term v_{k-1}, we construct the two following subproblems.

1. **Proximal point step.** We first perform an inexact proximal point step with prox-center x_{k-1}:

$$\bar{x}_k \approx \operatorname*{argmin}_x f_\kappa(x; x_{k-1}) \quad \text{[Proximal-point step]}$$

2. **Accelerated proximal point step.** Then we build the next prox-center y_k as the convex combination

$$y_k = \alpha_k v_{k-1} + (1 - \alpha_k) x_{k-1}. \tag{3.3}$$

Next, we use y_k as a prox-center and update the next extrapolation term:

$$\begin{aligned} &\tilde{x}_k \approx \operatorname*{argmin}_x f_\kappa(x; y_k) && \text{[Accelerated proximal-point step]} \\ &v_k = x_{k-1} + \tfrac{1}{\alpha_k}(\tilde{x}_k - x_{k-1}) && \text{[Extrapolation]} \end{aligned} \tag{3.4}$$

where $\alpha_{k+1} \in (0,1)$ is a sequence of coefficients satisfying $(1-\alpha_{k+1})/\alpha_{k+1}^2 = 1/{\alpha_k}^2$. Essentially, the sequences $(\alpha_k)_k, (y_k)_k, (v_k)_k$ are built upon the extrapolation principles of Nesterov [81].

Picking the best. At the end of iteration k, we have at hand two iterates, resp. $\bar{x}_k$ and $\tilde{x}_k$. Following [49], we simply choose the best of the two in terms of their objective values, that is we choose x_k such that

$$f(x_k) \leq \min\left\{f(\bar{x}_k), f(\tilde{x}_k)\right\}.$$

The proposed scheme blends the two steps in a synergistic way, allowing us to recover the near-optimal rates of convergence in both worlds: convex and non-convex. Intuitively, when $\bar{x}_k$ is chosen, it means that Nesterov's extrapolation step "fails" to accelerate convergence.

Stopping criterion for the subproblems. In order to derive complexity bounds, it is important to properly define the stopping criterion for the proximal subproblems. When the subproblem is convex, a functional gap like $f_\kappa(z;x) - \inf_z f_\kappa(z;x)$ may be used as a control of the inexactness, as in [69]. Without convexity, this criterion cannot be used since such quantities can not be easily bounded. In particular, first order methods seek points whose subgradient is small. Since small subgradients do not necessarily imply small function values in a non-convex setting, first order methods only test is for small subgradients. In contrast, in the convex setting, small subgradients imply small function values; thus a first order method in the convex setting can "test" for small function values. Hence, we cannot use a direct application of Catalyst [69] which uses the functional gap as a stopping criteria. Because we are working in the nonconvex setting, we include a stationarity stopping criteria.

```
Algorithm 12: 4WD-Catalyst
  input: Fix a point $x_0 \in \operatorname{dom} f$, real numbers $\kappa > 0$, and an optimization method $\mathcal{M}$.
  initialization: $\alpha_1 \equiv 1$, $v_0 \equiv x_0$.
  repeat for $k = 1, 2, \ldots$
    1. Choose $\bar{x}_k$ using $\mathcal{M}$ such that
         $$\bar{x}_k \approx \operatorname*{argmin}_x f_\kappa(x; x_{k-1}) \tag{3.5}$$
       where $\operatorname{dist}\big(0, \partial f_\kappa(\bar{x}_k; x_{k-1})\big) < \kappa \left\|\bar{x}_k - x_{k-1}\right\|$ and $f_\kappa(\bar{x}_k; x_{k-1}) \le f_\kappa(x_{k-1}; x_{k-1})$.
    2. Set
         $$y_k = \alpha_k v_{k-1} + (1 - \alpha_k) x_{k-1}. \tag{3.6}$$
    3. Choose $\tilde{x}_k$ using $\mathcal{M}$ such that
         $$\tilde{x}_k \approx \operatorname*{argmin}_x f_\kappa(x; y_k) \tag{3.7}$$
       where $\operatorname{dist}\big(0, \partial f_\kappa(\tilde{x}_k; y_k)\big) < \frac{\kappa}{k+1} \left\|\tilde{x}_k - y_k\right\|$
    4. Set
         $$v_k = x_{k-1} + \frac{1}{\alpha_k}(\tilde{x}_k - x_{k-1}). \tag{3.8}$$
    5. Pick $\alpha_{k+1} \in (0, 1)$ satisfying
         $$\frac{1 - \alpha_{k+1}}{\alpha_{k+1}^2} = \frac{1}{\alpha_k^2}. \tag{3.9}$$
    6. Choose $x_k$ to be any point satisfying
         $$f(x_k) \le \min\{f(\bar{x}_k), f(\tilde{x}_k)\}. \tag{3.10}$$
  until the stopping criterion $\operatorname{dist}\big(0, \partial f(\bar{x}_k)\big) < \varepsilon$
```

We propose to use jointly the following two types of stopping criteria:

1. Descent condition: $f_\kappa(z; y) \leq f_\kappa(y; y)$;

2. Adaptive stationary condition: $\operatorname{dist}\big(0, \partial f_\kappa(z; y)\big) < \kappa \left\| z - y \right\|$.

Without the descent condition, the stationarity condition is insufficient for defining a good stopping criterion because of the existence of local maxima in nonconvex problems. In the nonconvex setting, local maxima and local minima satisfy the stationarity condition. The descent condition ensures the iterates generated by the algorithm always decrease the value of objective function f; thus ensuring we move away from local maxima. The second criterion, adaptive stationary condition, provides a flexible relative tolerance on termination of algorithm used for solving the subproblems; a detailed analysis is forthcoming.

In 4WD-Catalyst , we use both the stationary condition and the descent condition as a stopping criteria to produce the point $\bar{x}$:

$$\operatorname{dist}\big(0, \partial f_\kappa(\bar{x}_k; x_{k-1})\big) < \kappa \left\| \bar{x}_k - x_{k-1} \right\| \text{ and } f_\kappa(\bar{x}_k; x_{k-1}) \leq f_\kappa(x_{k-1}; x_{k-1}). \tag{3.11}$$

For the point $\tilde{x}$, our “acceleration” point, we use a modified stationary condition:

$$\operatorname{dist}\big(0, \partial f_\kappa(\tilde{x}_k; y_k)\big) < \frac{\kappa}{k+1} \left\| \tilde{x}_k - y_k \right\|. \tag{3.12}$$

The $k+1$ factor guarantees 4WD-Catalyst accelerates for the convex setting. To be precise, Equation (B.5) in the proofs of Theorem 3.3.1 and Theorem 3.3.2 uses the factor $k+1$ to ensure convergence. Note, we do not need the descent condition for $\tilde{x}$, as the functional decrease in $\bar{x}$ is enough to ensure the sequence $\{f(x_k)\}_{k \geq 1}$ is monotonically decreasing.

3.3.2 Convergence analysis.

We present here the theoretical properties of Algorithm 12. In this first stage, we do not take into account the complexity of solving the subproblems (3.5) and (3.7). For the next two theorems, we assume that the stopping criteria for the proximal subproblems are satisfied at each iteration of Algorithm 12.

Theorem 3.3.1 (Outer-loop complexity for 4WD-Catalyst; non-convex case)**.** *For any $\kappa > 0$ and $N \geq 1$, the iterates generated by Algorithm 12 satisfy*

$$\min_{j=1,\ldots,N} \operatorname{dist}^2\big(0, \partial f(\bar{x}_j)\big) \leq \frac{8\kappa}{N}(f(x_0) - f^*).$$

It is important to notice that this convergence result is valid for any κ and does not require it to be larger than the weak convexity parameter. As long as the stopping criteria for the proximal subproblems are satisfied, the quantities $\operatorname{dist}(0, \partial f(\bar{x_j}))$ tend to zero. The proof is inspired by that of inexact proximal algorithms [6, 51, 69] and appears in Appendix B.2.

If the function f turns out to be convex, the scheme achieves a faster convergence rate both in function values and in stationarity:

Theorem 3.3.2 (Outer-loop complexity, convex case)**.** *If the function f is convex, then for any $\kappa > 0$ and $N \geq 1$, the iterates generated by Algorithm 12 satisfy*

$$f(x_N) - f(x^*) \leq \frac{4\kappa}{(N+1)^2} \left\|x^* - x_0\right\|^2, \tag{3.13}$$

and

$$\min_{j=1,\ldots,2N} \operatorname{dist}^2\big(0, \partial f(\bar{x}_j)\big) \leq \frac{32\kappa^2}{N(N+1)^2} \left\|x^* - x_0\right\|^2,$$

where x^* is any minimizer of the function f.

The proof of Theorem 3.3.2 appears in Appendix B.2. This theorem establishes a rate of $O(N^{-2})$ for suboptimality in function value and convergence in $O(N^{-3/2})$ for the minimal norm of subgradients. The first rate is optimal in terms of information-based complexity for the minimization of a convex composite function [81, 86]. The second can be improved to $O(N^{-2}\log(N))$ through a regularization technique, if one knew in advance that the function is convex and had an estimate on the distance of the initial point to an optimal solution [85].

Towards an automatically adaptive algorithm. So far, our analysis has not taken into account the cost of obtaining the iterates $\bar{x}_j$ and $\tilde{x}_j$ by the algorithm $\mathcal{M}$. We emphasize again that the two results above do not require any assumption on κ, which leaves us a degree of freedom. In order to develop the global complexity, we need to evaluate the total number of iterations performed by $\mathcal{M}$ throughout the process. Clearly, this complexity heavily depends on the choice of κ, since it controls the magnitude of regularization we add to improve the convexity of the subproblem. This is the point where a careful analysis is needed, because our algorithm must adapt to ρ without knowing it in advance. The next section is entirely dedicated to this issue. In particular, we will explain how to automatically adapt the parameter κ (Algorithm 13).

3.4 The 4WD-Catalyst-Automatic algorithm

In this section, we work towards understanding the global efficiency of Algorithm 12, which automatically adapts to the weak convexity parameter. For this, we must take into account the cost of approximately solving the proximal subproblems to the desired stopping criteria. We expect that once the subproblem becomes strongly convex, the given optimization method $\mathcal{M}$ can solve it efficiently. For this reason, we first focus on the computational cost for solving the sub-problems, before introducing a new algorithm with known worst-case complexity.

3.4.1 Solving the sub-problems efficiently

When κ is large enough, the subproblems become strongly convex; thus globally solvable. Henceforth, we will assume that $\mathcal{M}$ satisfies the following natural linear convergence assumption.

Linear convergence of $\mathcal{M}$ for strongly-convex problems. We assume that for any $\kappa > \rho$, there exist $A_\kappa \geq 0$ and $\tau_\kappa \in (0,1)$ so that the following hold:

1. For any prox-center $y \in \mathbb{R}^p$ and initial $z_0 \in \mathbb{R}^p$ the iterates $\{z_t\}_{t\geq 1}$ generated by $\mathcal{M}$ on the problem $\min_z f_\kappa(z;y)$ satisfy

$$\mathrm{dist}^2(0, \partial f_\kappa(z_t;y)) \leq A_\kappa(1-\tau_\kappa)^t(f_\kappa(z_0;y) - f_\kappa^*(y)), \tag{3.14}$$

where $f_\kappa(y)^* := \inf_z f_\kappa(z; y)$. If the method $\mathcal{M}$ is randomized, we require the same inequality to hold in expectation.

2. The rates τ_κ and the constants A_κ are increasing in κ.

Remark 5. The linear convergence we assume here for $\mathcal{M}$ differs from the one considered by [69], which was given in terms of function values. However, if the problem is a composite one, both points of view are near-equivalent, as discussed in Section B.1 and the precise relationship is given in Appendix B.3. We choose the norm of the subgradient as our measurement because the complexity analysis is easier.

Then, a straightforward analysis bounds the computational complexity to achieve an ε-stationary point.

Lemma 3.4.1. *Let us consider a strongly convex problem $f_\kappa(\cdot; y)$ and a linearly convergent method $\mathcal{M}$ generating a sequence of iterates $\{z_t\}_{t\geq 0}$. Define*
$T(\varepsilon) = \inf\{t \geq 1, \operatorname{dist}\big(0, \partial f_\kappa(z_t; y)\big) \leq \varepsilon\}$, *where ε is the target accuracy; then,*

1. *If $\mathcal{M}$ is deterministic,*

$$T(\varepsilon) \leq \frac{1}{\tau_\kappa} \log\left(\frac{A_\kappa\left(f_\kappa(z_0; y) - f_\kappa^*(y)\right)}{\varepsilon^2}\right).$$

2. *If $\mathcal{M}$ is randomized, then*

$$\mathbb{E}\left[T(\varepsilon)\right] \leq \frac{1}{\tau_\kappa} \log\left(\frac{A_\kappa\left(f_\kappa(z_0; y) - f_\kappa^*(y)\right)}{\tau_\kappa \varepsilon^2}\right).$$

see Lemma C.1 of [69].

As we can see, we only lose a factor in the log term by switching from deterministic to randomized algorithms. For the sake of simplicity, we perform our analysis only for deterministic algorithms and the analysis for randomized algorithms holds in the same way in expectation.

Bounding the required iterations when $\kappa > \rho$ and restart strategy. Recall that we add a quadratic to f with the hope to make each subproblem convex. Thus, if ρ is known, then we should set $\kappa > \rho$. In this first stage, we show that whenever $\kappa > \rho$, then the number of inner calls to $\mathcal{M}$ can be bounded with a proper initialization. Consider the subproblem

$$\min_{x \in \mathbb{R}^p} \left\{ f_\kappa(x; y) = f(x) + \frac{\kappa}{2} \|x - y\|^2 \right\}, \tag{3.15}$$

and define the initialization point z_0 by

1. if f is smooth, then set $z_0 = y$;

2. if $f = f_0 + \psi$ is composite, with f_0 L-smooth, then set $z_0 = \text{prox}_{\eta\psi}(y - \eta \nabla f_0(y))$ with $\eta \leq \frac{1}{L+\kappa}$.

Theorem 3.4.2. *Consider the subproblem (3.15) and suppose $\kappa > \rho$. Then initializing $\mathcal{M}$ at the previous z_0 generates a sequence of iterates $(z_t)_{t \geq 0}$ such that*

1. *in at most T_κ iterations where*

$$T_\kappa = \frac{1}{\tau_\kappa} \log \left(\frac{8A_\kappa(L+\kappa)}{(\kappa - \rho)^2} \right),$$

the output z_T satisfies $f_\kappa(z_T; y) \leq f_\kappa(z_0; y)$ (descent condition) and dist$(0, \partial f_\kappa(z_T; y)) \leq \kappa \|z_T - y\|$ (adaptive stationary condition);

2. *in at most $S_\kappa \log(k+1)$ iterations where*

$$S_\kappa \log(k+1) = \frac{1}{\tau_\kappa} \log \left(\frac{8A_\kappa(L+\kappa)(k+1)^2}{(\kappa - \rho)^2} \right),$$

the output z_S satisfies dist$(0, \partial f_\kappa(z_S; y)) \leq \frac{\kappa}{k+1} \|z_S - y\|$ (modified adaptive stationary condition).

The proof is technical and is presented in Appendix B.4. The lesson we learn here is that as soon as the subproblem becomes strongly convex, it can be solved in almost a constant number of iterations. Herein arises a problem–the choice of the smoothing parameter κ. On one hand, when f is already convex, we may want to choose κ small in order to obtain the desired optimal complexity. On the other hand, when the problem is non convex, a small κ may not ensure the strong convexity of the subproblems. Because of such different behavior according to the convexity of the function, we introduce an additional parameter κ_{cvx} to handle the regularization of the extrapolation step. Moreover, in order to choose a $\kappa > \rho$ in the nonconvex case, we need to know in advance an estimate of ρ. This is not an easy task for large scale machine learning problems such as neural networks. Thus we propose an adaptive step to handle it automatically.

3.4.2 4WD-Catalyst-Automatic: adaptation to weak convexity

We now introduce 4WD-Catalyst-Automatic, presented in Algorithm 13, which can automatically adapt to the *unknown weak convexity* constant of the objective. The algorithm relies on a procedure to automatically adapt to ρ, described in Algorithm 14.

The idea is to fix in advance a number of iterations T, let $\mathcal{M}$ run on the subproblem for T iterations, output the point z_T, and check if a sufficient decrease occurs. We show that if we set $T = \widetilde{O}(\tau_L^{-1})$, where the notation $\widetilde{O}$ hides logarithmic dependencies in L and A_L, where L is the Lipschitz constant of the smooth part of f; then, if the subproblem were convex, the following conditions would be guaranteed:

1. Descent condition: $f_\kappa(z_T; x) \leq f_\kappa(x; x)$;

2. Adaptive stationary condition: $\text{dist}\big(0, \partial f_\kappa(z_T; x)\big) \leq \kappa \left\| z_T - x \right\|$.

Thus, if either condition is not satisfied, then the subproblem is deemed not convex and we double κ and repeat. The procedure yields an estimate of ρ in a logarithmic number of increases; see Lemma B.4.3.

Relative stationarity and predefining S. One of the main differences of our approach with the Catalyst algorithm of [69] is to use a *pre-defined* number of iterations, T and S,

```
Algorithm 13: 4WD-Catalyst-Automatic
input: Fix a point $x_0 \in \operatorname{dom} f$, real numbers $\kappa_0, \kappa_{\text{cvx}} > 0$ and $T, S > 0$, and an optimization method $\mathcal{M}$.
initialization: $\alpha_1 = 1$, $v_0 = x_0$.
repeat for $k = 1, 2, \ldots$
    1. Compute
           $(\bar{x}_k, \kappa_k) = \textsf{Auto-adapt}\ (x_{k-1}, \kappa_{k-1}, T).$
    2. Compute $y_k = \alpha_k v_{k-1} + (1 - \alpha_k) x_{k-1}$ and apply $S \log(k+1)$ iterations of $\mathcal{M}$ to find
           $\tilde{x}_k \approx \operatorname*{argmin}_{x \in \mathbb{R}^p} f_{\kappa_{\text{cvx}}}(x, y_k),$    (3.16)
       by using the initialization strategy described below (3.15).
    3. Update $v_k$ and $\alpha_{k+1}$ by
           $v_k = x_{k-1} + \frac{1}{\alpha_k}(\tilde{x}_k - x_{k-1}) \quad \text{and} \quad \alpha_{k+1} = \frac{\sqrt{\alpha_k^4 + 4\alpha_k^2} - \alpha_k^2}{2}.$
    4. Choose $x_k$ to be any point satisfying $f(x_k) = \min\{f(\bar{x}_k), f(\tilde{x}_k)\}$.
until the stopping criterion $\operatorname{dist}\big(0, \partial f(\bar{x}_k)\big) < \varepsilon$
```

for solving the subproblems. We introduce κ_{cvx}, a $\mathcal{M}$ dependent smoothing parameter and set it in the same way as the smoothing parameter in [69]. The automatic acceleration of our algorithm when the problem is convex is due to extrapolation steps in Step 2-3 of 4WD-Catalyst. We show that if we set $S = \widetilde{O}\left(\tau_{\kappa_{\text{cvx}}}^{-1}\right)$, where $\widetilde{O}$ hides logarithmic dependencies in L, κ_κ, and $A_{\kappa_{\text{cvx}}}$, then we can be sure that, for convex objectives,

$$\operatorname{dist}\big(0, \partial f_{\kappa_{\text{cvx}}}(\tilde{x}_k; y_k)\big) < \frac{\kappa_{\text{cvx}}}{k+1} \left\| \tilde{x}_k - y_k \right\|. \tag{3.17}$$

This relative stationarity of $\tilde{x}_k$, including the choice of κ_{cvx}, shall be crucial to guarantee that the scheme accelerates in the convex setting. An additional $k+1$ factor appears compared to the previous adaptive stationary condition because we need higher accuracy for solving the subproblem to achieve the accelerated rate in $1/\sqrt{\varepsilon}$.

We shall see in the experiments that our strategy of predefining T and S works quite well. The theoretical bounds we derive are, in general, too conservative; we observe in our experiments that one may choose T and S significantly smaller than the theory suggests and still retain the stopping criteria.

Algorithm 14: Auto-adapt (y, κ, T)

```
input: $y \in \mathbb{R}^p$, method $\mathcal{M}$, $\kappa > 0$, number of iterations $T$.
Repeat Compute
    $$z_T \approx \operatorname*{argmin}_{z \in \mathbb{R}^p} f_\kappa(z; y).$$
  by running $T$ iterations of $\mathcal{M}$ by using the initialization strategy described
  below (3.15).
If $f_\kappa(z_T; y) > f_\kappa(y; y)$ or $\text{dist}(\partial f_\kappa(z_T; y), 0) > \kappa \|z_T - y\|$,
then go to repeat with $\kappa \to 2\kappa$.
else go to output.
output $(z_T, \kappa)$.
```

To derive the global complexity results for 4WD-Catalyst-Automatic that match optimal convergence guarantees, we make a distinction between the regularization parameter κ in the proximal point step and in the extrapolation step. For the proximal point step, we apply Algorithm 14 to adaptively produce a sequence of κ_k initializing at $\kappa_0 > 0$, an initial guess of ρ. The resulting $\bar{x}_k$ and κ_k satisfy both the following inequalities:

$$\text{dist}\big(0, \partial f_{\kappa_k}(\bar{x}_k; x_{k-1})\big) < \kappa_k \|\bar{x}_k - x_k\| \text{ and } f_{\kappa_k}(\bar{x}_k; x_{k-1}) \leq f_{\kappa_k}(x_{k-1}; x_{k-1}). \tag{3.18}$$

For the extrapolation step, we introduce the parameter κ_{cvx} which essentially depends on the Lipschitz constant L. The choice is the same as the smoothing parameter in [69] and depends on the method $\mathcal{M}$. With a similar predefined iteration strategy, the resulting $\tilde{x}_k$

satisfies the following inequality if the original objective is convex,

$$\operatorname{dist}\big(0, \partial f_{\kappa_{\text{cvx}}}(\tilde{x}_k; y_k)\big) < \frac{\kappa_{\text{cvx}}}{k+1} \|\tilde{x}_k - y_k\| . \tag{3.19}$$

3.4.3 Convergence analysis

Let us next postulate that T and S are chosen large enough to guarantee that $\bar{x}_k$ and $\tilde{x}_k$ satisfy conditions (3.18) and (3.19) for the corresponding subproblems, and see how the outer algorithm complexity resembles the guarantees of Theorem 3.3.1 and Theorem 3.3.2. The main technical difference is that κ changes at each iteration k, which requires keeping track of the effects of κ_k and κ_{cvx} on the proof.

Theorem 3.4.3 (Outer-loop complexity, 4WD-Catalyst-Automatic). *Fix real constants $\kappa_0, \kappa_{cvx} > 0$, and $x_0 \in dom\ f$. Set $\kappa_{\max} := \max_{k\geq 1} \kappa_k$. Suppose that the number of iterations T is such that $\bar{x}_k$ satisfies* (3.18)*. Define $f^* := \lim_{k\to\infty} f(x_k)$. Then for any $N \geq 1$, the iterates generated by Algorithm 13 satisfy,*

$$\min_{j=1,\dots,N} \operatorname{dist}^2\big(0, \partial f(\bar{x}_j)\big) \leq \frac{8\kappa_{\max}}{N}(f(x_0) - f^*).$$

If in addition the function f is convex and S_k is chosen so that $\tilde{x}_k$ satisfies (3.19)*, then*

$$\min_{j=1,\dots,2N} \operatorname{dist}^2\big(0, \partial f(\bar{x}_j)\big) \leq \frac{32\kappa_{\max}\kappa_{cvx}}{N(N+1)^2} \|x^* - x_0\|^2 ,$$

and

$$f(x_N) - f(x^*) \leq \frac{4\kappa_{cvx}}{(N+1)^2} \|x^* - x_0\|^2 , \tag{3.20}$$

where x^* is any minimizer of the function f.

Inner-loop Complexity In light of Theorem 3.4.3, we must now understand how to choose T and S as small as possible, while guaranteeing that $\bar{x}_k$ and $\tilde{x}_k$ satisfy (3.18) and (3.19) hold for each k. The quantities T and S depend on the method $\mathcal{M}$'s convergence rate parameter τ_κ which only depends on L and κ. For example, the convergence rate parameter $\tau_\kappa^{-1} = (L+\kappa)/\kappa$ for gradient descent and $\tau_\kappa^{-1} = n+(L+\kappa)/\kappa$ for SVRG. The values of T and

S must be set beforehand without knowing the true value of the weak convexity constant ρ. Using Theorem 3.4.2, we assert the following choices for T and S.

Theorem 3.4.4 (Inner complexity for 4WD-Catalyst-Automatic : determining the values T and S)**.** *Suppose the stopping criteria are* (3.18) *and* (3.19) *as in in Theorem 3.4.3, and choose T and S in Algorithm 13 to be the smallest numbers satisfying*

$$T \geq \frac{1}{\tau_L} \log\left(\frac{40 A_{4L}}{L}\right),$$

and

$$S \log(k+1) \geq \frac{1}{\tau_{\kappa_{cvx}}} \log\left(\frac{8 A_{\kappa_{cvx}} (\kappa_{cvx} + L)(k+1)^2}{\kappa_{cvx}^2}\right),$$

for all k. In particular,

$$T = O\left(\frac{1}{\tau_L} \log\left(A_{4L}, L\right)\right),$$
$$S = O\left(\frac{1}{\tau_{\kappa_{cvx}}} \log(A_{\kappa_{cvx}}, L, \kappa_{cvx})\right).$$

Then $\kappa_{\max} \leq 4L$ and the following hold for any index $k \geq 1$:

1. *Generating $\bar{x}_k$ in Algorithm 13 requires at most $\widetilde{O}\left(\tau_L^{-1}\right)$ iterations of $\mathcal{M}$;*

2. *Generating $\tilde{x}_k$ in Algorithm 13 requires at most $\widetilde{O}\left(\tau_{\kappa_{cvx}}^{-1}\right)$ iterations of $\mathcal{M}$.*

where $\widetilde{O}$ hides universal constants and logarithmic dependencies on k, L, κ_{cvx}, A_L, and $A_{\kappa_{cvx}}$.

Appendix B.4 is devoted to proving Theorem 3.4.4, but we outline below the general procedure and state the two main propositions (see Proposition 3.4.5 and Proposition 3.4.6).

We summarize the proof of Theorem 3.4.4 as followed:

1. When $\kappa > \rho + L$, we compute the number of iterations of $\mathcal{M}$ to produce a point satisfying (3.18). Such a point will become $\bar{x}_k$.

2. When the function f is convex, we compute the number of iterations of $\mathcal{M}$ to produce a point which satisfies the (3.19) condition. Such a point will become the point $\tilde{x}_k$.

3. We compute the smallest number of times we must double κ_0 until it becomes larger than $\rho + L$. Thus eventually the condition $4L \geq \kappa > \rho + L$ will occur.

4. We always set the number of iterations of $\mathcal{M}$ to produce $\bar{x}_k$ and $\tilde{x}_k$ as in Step 1 and Step 2, respectively, regardless of whether $f_\kappa(\cdot; x_k)$ is convex or f is convex.

The next proposition shows that Auto-adapt terminates with a suitable choice for $\bar{x}_k$ after T number of iterations.

Proposition 3.4.5 (Inner complexity for $\bar{x}_k$)**.** *Suppose $\rho + L < \kappa \leq 4L$. By initializing the method $\mathcal{M}$ using the strategy suggested in Algorithm 13 for solving*

$$\min_z \left\{ f_\kappa(z; x) := f(z) + \frac{\kappa}{2} \|z - x\|^2 \right\}$$

we may run the method $\mathcal{M}$ for at least T iterations, where

$$T \geq \frac{1}{\tau_L} \log \left(\frac{40 A_{4L}}{L} \right);$$

then, the output z_T satisfies $f_\kappa(z_T; x) \leq f_\kappa(x; x)$ and dist$\big(0, \partial f_\kappa(z_T; x)\big) \leq \kappa \|z_T - x\|$.

Under the additional assumption that the function f is convex, we produce a point with (3.19) when the number of iterations S is chosen sufficiently large.

Proposition 3.4.6 (Inner-loop complexity for $\tilde{x}_k$)**.** *Consider the method $\mathcal{M}$ with the initialization strategy suggested in Algorithm 13 for minimizing $f_{\kappa_{cvx}}(\cdot; y_k)$ with linear convergence rates of the form* (3.14)*. Suppose the function f is convex. If the number of iterations of $\mathcal{M}$ is greater than*

$$S = O\left(\frac{1}{\tau_{\kappa_{cvx}}} \log(A_{\kappa_{cvx}}, L, \kappa_{cvx}) \right)$$

such that

$$S \log(k+1) \geq \frac{1}{\tau_{\kappa_{cvx}}} \log \left(\frac{8 A_{\kappa_{cvx}} (\kappa_{cvx} + L)(k+1)^2}{\kappa_{cvx}^2} \right), \tag{3.21}$$

then, the output $\tilde{z}_S = \tilde{x}_k$ satisfies $\|\partial f_{\kappa_{cvx}}(\tilde{z}_S)\| < \frac{\kappa_{cvx}}{k+1} \|\tilde{z}_{S_k} - y_k\|$ for all $k \geq 1$.

We can now derive global complexity bounds by combining Theorem 3.4.3 and Theorem 3.4.4, and a good choice for the constant κ_{cvx}.

Theorem 3.4.7 (Global complexity bounds for 4WD-Catalyst-Automatic)**.** *Choose Choose T and S as in Theorem 3.4.4. We let $\widetilde{O}$ hide universal constants and logarithmic dependencies in A_L, $A_{\kappa_{cvx}}$, L, ε, κ_0, κ_{cvx}, and $\|x^* - x_0\|^2$. Then, the following statements hold.*

1. *Algorithm 13 generates a point x satisfying dist$\big(0, \partial f(x)\big) \leq \varepsilon$ after at most*
$$\widetilde{O}\left(\left(\tau_L^{-1} + \tau_{\kappa_{cvx}}^{-1}\right) \cdot \frac{L(f(x_0) - f^*)}{\varepsilon^2}\right)$$
iterations of the method $\mathcal{M}$.

2. *If f is convex, then Algorithm 13 generates a point x satisfying dist$\big(0, \partial f(x)\big) \leq \varepsilon$ after at most*
$$\widetilde{O}\left(\left(\tau_L^{-1} + \tau_{\kappa_{cvx}}^{-1}\right) \cdot \frac{L^{1/3}\left(\kappa_{cvx}\|x^* - x_0\|^2\right)^{1/3}}{\varepsilon^{2/3}}\right)$$
iterations of the method $\mathcal{M}$.

3. *If f is convex, then Algorithm 13 generates a point x satisfying $f(x) - f^* \leq \varepsilon$ after at most*
$$\widetilde{O}\left(\left(\tau_L^{-1} + \tau_{\kappa_{cvx}}^{-1}\right) \cdot \frac{\sqrt{\kappa_{cvx}\|x^* - x_0\|^2}}{\sqrt{\varepsilon}}\right)$$
iterations of the method $\mathcal{M}$.

Remark 6. In general, the linear convergence parameter of $\mathcal{M}$, τ_κ, depends on the condition number of the problem f_κ. Here, τ_L and $\tau_{\kappa_{\text{cvx}}}$ are precisely given by plugging in $\kappa = L$ and κ_{cvx} respectively into τ_κ. To clarify, let $\mathcal{M}$ be SVRG, τ_κ is given by $\frac{1}{n + \frac{\kappa + L}{\kappa}}$ which yields $\tau_L = 1/(n+2)$. A more detailed computation is given in Table 3.5.1. For all the incremental methods we considered, these parameters τ_L and τ_κ are on the order of $1/n$.

Remark 7. If $\mathcal{M}$ is a first order method, the convergence guarantee in the convex setting is *near-optimal*, up to logarithmic factors, when compared to $O(1/\sqrt{\varepsilon})$ [69, 114]. In the non-convex setting, our approach matches, up to logarithmic factors, the best known rate for this class of functions, namely $O(1/\varepsilon^2)$ [21, 20]. Moreover, our rates dependence on the dimension and Lipschitz constant equals, up to log factors, the best known dependencies in both the convex and nonconvex setting. These logarithmic factors may be the price we pay for having a generic algorithm.

3.5 Applications to Existing Algorithms

We now show how to accelerate existing algorithms $\mathcal{M}$ and compare the convergence guaranties before and after 4WD-Catalyst-Automatic. In particular, we focus on the gradient descent algorithm and on the incremental methods SAGA and SVRG. For all the algorithms considered, we state the convergence guaranties in terms of the *total number of iterations* (in expectation, if appropriate) to reach an accuracy of ε; in the convex setting, the accuracy is stated in terms of functional error, $f(x) - \inf f < \varepsilon$ and in the nonconvex setting, the appropriate measure is stationarity, namely $\mathrm{dist}(0, \partial f(x)) < \varepsilon$. All the algorithms considered have formulations for the composite setting with analogous convergence rates. Table 3.5 presents convergence rates for SAGA [34], (prox) SVRG [116], and gradient descent (FG).

The original SVRG [116] has no guarantees for nonconvex functions; however, there is a nonconvex extension of SVRG in [94]. Their convergence rate achieves a better dependance on n compared to our results, namely $O(\frac{n^{2/3}L}{\varepsilon^2})$. This is done by performing a strategy of minibatching. In order to achieve a similar dependency on n, we require a tighter bound for SVRG with minibatching applied to μ-strongly convex problems, namely $O\left(\left(n^{2/3} + \frac{L}{\mu}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$. To the best of our knowledge, such a rate is currently unknown.

3.5.1 Practical parameters choices and convergence rates

The smoothing parameter κ_{cvx} drives the convergence rate of 4WD-Catalyst-Automatic in the convex setting. To determine κ_{cvx}, we pretend $\rho = 0$ and compute the global complexity of our scheme. As such, we end up with the same complexity result as Catalyst [69]. Following their work, the rule of thumb is to maximize the ratio $\tau_\kappa/\sqrt{L+\kappa}$ for convex problems. On

	Nonconvex		Convex	
	Original	4WD-Catalyst-Auto.	Original	4WD-Catalyst-Auto.
FG	$\mathcal{O}\left(n\frac{L}{\varepsilon^2}\right)$	$\widetilde{O}\left(n\frac{L}{\varepsilon^2}\right)$	$O\left(n\frac{L}{\varepsilon}\right)$	$\widetilde{O}\left(n\sqrt{\frac{L}{\varepsilon}}\right)$
SVRG [116]	not avail.	$\widetilde{O}\left(n\frac{L}{\varepsilon^2}\right)$	not avail.	$\widetilde{O}\left(\sqrt{n}\sqrt{\frac{L}{\varepsilon}}\right)$
SAGA [34]	not avail.	$\widetilde{O}\left(n\frac{L}{\varepsilon^2}\right)$	$O\left(n\frac{L}{\varepsilon}\right)$	$\widetilde{O}\left(\sqrt{n}\sqrt{\frac{L}{\varepsilon}}\right)$

Table 3.1: Comparison of rates of convergence, before and after the 4WD-Catalyst-Automatic , resp. in the non-convex and convex cases. For the comparision, in the convex case, we only present the number of iterations to obtain a point x satisfying $f(x) - f^* < \varepsilon$. In the non-convex case, we show the number of iterations to obtain a point x satisfying $\text{dist}(0, \partial f(x)) < \varepsilon$.

the other hand, the choice of κ_0 is independent of $\mathcal{M}$; it is an initial lower estimate for the weak convexity constant ρ. In practice, we typically choose $\kappa_0 = \kappa_{\text{cvx}}$; For incremental approaches a natural heuristic is also to choose $S = T = n$, meaning that S iterations of $\mathcal{M}$ performs one pass over the data. In Table 3.5.1, we present the values of κ_{cvx} used for various algorithms, as well as other quantities that are useful to derive the convergence rates.

Full gradient method. A first illustration is the algorithm obtained when accelerating the regular "full" gradient (FG). Here, the optimal choice for κ_{cvx} is L. In the convex setting, we get an accelerated rate of $O(n\sqrt{L/\varepsilon}\log(1/\varepsilon))$ which agrees with Nesterov's accelerated variant (AFG) up to logarithmic factors. On the other hand, in the nonconvex setting, our approach achieves no worse rate than $O(nL/\varepsilon^2\log(1/\varepsilon))$, which agrees with the standard gradient descent up to logarithmic factors. We note that under stronger assumptions, namely C^2-smoothness of the objective, the accelerated algorithm in [19] achieves the same rate as (AFG) for the convex setting and $O(\varepsilon^{-7/4}\log(1/\varepsilon))$ for the nonconvex setting. Their approach, however, does not extend to composite setting nor to stochastic methods. Our marginal loss is the price we pay for considering a much larger class of functions.

Randomized incremental gradient. We now consider randomized incremental gradient methods such as SAGA [34] and (prox) SVRG [116]. Here, the optimal choice for κ_{cvx} is $O(L/n)$. Under the convex setting, we achieve an accelerated rate of $O(\sqrt{n}\sqrt{L/\varepsilon}\log(1/\varepsilon))$. A direct application of SVRG and SAGA have no convergence guarantees in the non-convex setting. With our approach, the resulting algorithm matches the guarantees for FG up to log factors.

Variable	Description	GD	SVRG	SAGA
$1/\tau_L$	linear convergence parameter with $\kappa = L$	2	$n+2$	$4n$
κ_{cvx}	smoothing parameter for convex setting	L	$L/(n-1)$	$3L/(4n-3)$
$1/\tau_{\kappa_{\text{cvx}}}$	linear convergence parameter with κ_{cvx}	2	$2n$	$4n$
A_{4L}	constant from the convergence rate of $\mathcal{M}$	$8L$	$8L$	$8Ln$

Table 3.2: Values of various quantities that are useful to derive the convergence rate of the different optimization methods.

3.5.2 Detailed derivation of convergence rates

Using the values of Table 3.5.1, we may now specialize our convergence results to different methods.

Gradient descent. The number of iterations in the inner loop are

$$T \geq 2\log(320)$$
$$S\log(k+1) \geq 2\log\left(64(k+1)^2\right)$$

The global complexity for gradient descent is

1. Algorithm 13 will generate a point x satisfying $\operatorname{dist}\big(0, \partial f(x)\big) \leq \varepsilon$ after at most
$$O\left[\frac{nL(f(x_0)-f^*)}{\varepsilon^2} \cdot \log\left(\frac{L^2(f(x_0)-f^*)^2}{\varepsilon^4}\right) + n\log\left(\frac{L}{\kappa_0}\right)\right]$$
gradient computations.

2. If f is convex, then Algorithm 13 will generate a point x satisfying $\operatorname{dist}\big(0, \partial f(x)\big) \leq \varepsilon$ after at most
$$O\left[\frac{nL^{2/3}\left\|x_0-x^*\right\|^{2/3}}{\varepsilon^{2/3}} \cdot \log\left(\frac{L^{4/3}\left\|x_0-x^*\right\|^{4/3}}{\varepsilon^{4/3}}\right) + n\log\left(\frac{L}{\kappa_0}\right)\right]$$
gradient computations.

3. If f is convex, then Algorithm 13 will generate a point x satisfying $f(x)-f^* \leq \varepsilon$ after at most
$$O\left[\frac{n\sqrt{L}\left\|x^*-x_0\right\|}{\sqrt{\varepsilon}} \cdot \log\left(\frac{L\left\|x_0-x^*\right\|^2}{\varepsilon}\right) + n\log\left(\frac{L}{\kappa_0}\right)\right]$$
gradient computations.

SVRG. For SVRG, the number of iterations in the inner loop are

$$T \geq (n+2)\log(320)$$
$$S\log(k+1) \geq 2n\log\left(64 \cdot n^2 \cdot (k+1)^2\right).$$

The global complexity for SVRG when n is sufficiently large is

1. Algorithm 13 will generate a point x satisfying $\operatorname{dist}\big(0, \partial f(x)\big) \leq \varepsilon$ after at most
$$O\left[\frac{nL(f(x_0) - f^*)}{\varepsilon^2} \cdot \log\left(\frac{n^2L^2(f(x_0) - f^*)^2}{\varepsilon^4}\right) + n\log\left(\frac{L}{\kappa_0}\right)\right]$$
gradient computations.

2. If f is convex, then Algorithm 13 will generate a point x satisfying $\operatorname{dist}\big(0, \partial f(x)\big) \leq \varepsilon$ after at most
$$O\left[\frac{n^{2/3}L^{2/3}\left\|x^* - x_0\right\|^{2/3}}{\varepsilon^{2/3}} \log\left(\frac{n^{4/3}L^{4/3}\left\|x^* - x_0\right\|^{4/3}}{\varepsilon^{4/3}}\right) + n^{2/3}\log\left(\frac{L}{\kappa_0}\right)\right]$$
gradient computations.

3. If f is convex, then Algorithm 13 will generate a point x satisfying $f(x) - f^* \leq \varepsilon$ after at most
$$O\left[\frac{\sqrt{nL}\left\|x^* - x_0\right\|}{\sqrt{\varepsilon}} \cdot \log\left(\frac{nL\left\|x_0 - x^*\right\|^2}{\varepsilon}\right) + \sqrt{n}\log\left(\frac{L}{\kappa_0}\right)\right]$$
gradient computations.

SAGA We observe that the variables for SAGA are the same as for SVRG up to a multiplicative factors. Therefore, the global complexities results for SAGA are, up to constant factors, the same as SVRG.

3.6 Experiments

We investigate the performance of 4WD-Catalyst-Automatic in two standard non-convex problems in machine learning. We report experimental results of 4WD-Catalyst-Automatic when applied to two different algorithms: SVRG [116] and SAGA [34]. We compare the following algorithms:

- Nonconvex SVRG/SAGA [94]: stepsize $\eta = 1/Ln^{2/3}$;

- Convex SVRG/SAGA [34, 116]: stepsize $\eta = 1/2L$;

- 4WD-Catalyst SVRG/SAGA: stepsize $\eta = 1/2L$.

The original version of SVRG (resp. SAGA), convex SVRG (resp. SAGA), was designed for minimizing convex objectives. We report their results, while there is no theoretical guarantee on their behavior when venturing into nonconvex terrains. We also report the results of recently proposed variants, Nonconvex SVRG/SAGA, designed for minimizing nonconvex objectives. The proposed algorithms 4WD-Catalyst SVRG and 4WD-Catalyst SAGA enjoy the strong theoretical guarantees stated in Sec. 3.

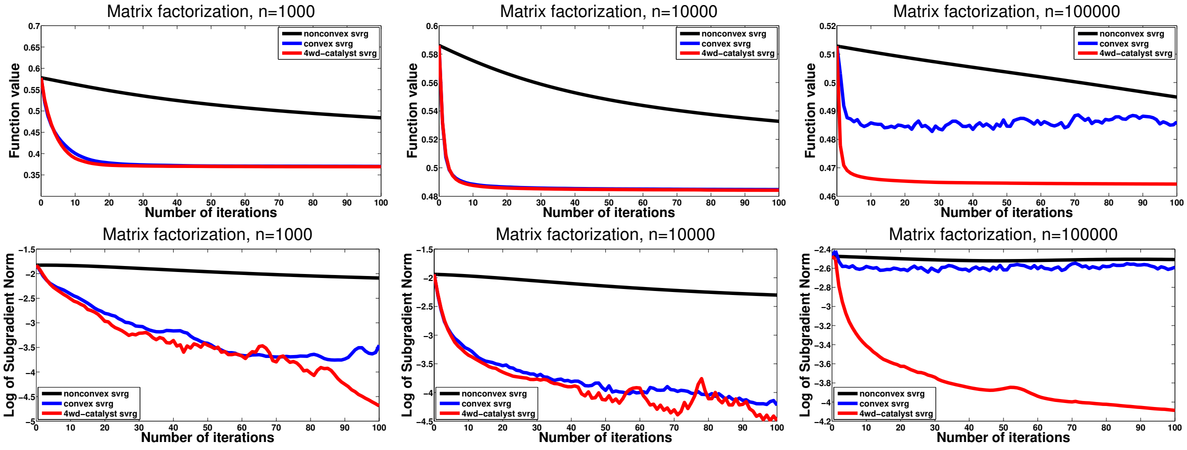
Parameter settings We start from an initial estimate of the Lipschitz constant L and use the theoretically recommended $\kappa_0 = \kappa_{\text{cvx}} = 2L/n$. The number of inner iterations is to $T = S = n$ in all experiments, which boils down to making one pass at most over the data for solving each sub-problem. We simply drop the $\log(k)$ dependency while solving the subproblem in (3.16). These choices turn out to be justified *a posteriori*, as both SVRG and SAGA have a much better convergence rate in practice than the theoretical rate derived from a worst-case analysis. Indeed, in all experiments, one pass over the data to solve each sub-problem is enough to guarantee sufficient descent.

Sparse matrix factorization a.k.a. dictionary learning. Dictionary learning consists of representing a dataset $X = [x_1, \cdots, x_n] \in \mathbb{R}^{m \times n}$ as a product $X \approx DA$, where D in $\mathbb{R}^{m \times p}$ is called a dictionary, and A in $\mathbb{R}^{p \times n}$ is a sparse matrix. The classical non-convex formulation [see 72] is

$$\min_{D \in \mathcal{C}, A \in \mathbb{R}^{p \times n}} \sum_{i=1}^{n} \frac{1}{2} \|x_i - D\alpha_i\|_2^2 + \psi(\alpha_i),$$

where $A = [\alpha_1 \cdots \alpha_n]$ carries the decomposition coefficients of signals $x_1 \cdots x_n$, ψ is a sparsity-inducing regularization and $\mathcal{C}$ is chosen as the set of matrices whose columns are in the ℓ_2-ball. An equivalent point of view is the finite-sum problem $\min_{D \in \mathcal{C}} \frac{1}{n} \sum_{i=1}^{n} f_i(D)$ with

$$f_i(D) := \min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \|x_i - D\alpha\|_2^2 + \psi(\alpha). \tag{3.22}$$



Matrix factorization, n=1000

Matrix factorization, n=10000

Matrix factorization, n=100000

Matrix factorization, n=1000

Matrix factorization, n=10000

Matrix factorization, n=100000

Figure 3.1: Dictionary learning experiments using SVRG. We plot the function value (top) and the subgradient norm (bottom). From left to right, we vary the size of dataset from $n = 1\,000$ to $n = 100\,000$.

We consider the elastic-net regularization $\psi(\alpha) = \frac{\mu}{2}\|\alpha\|^2 + \lambda\|\alpha\|_1$ of [119], which has a sparsity-inducing effect, and report the corresponding results in Figures 3.1 and 3.2, learning a dictionary in $\mathbb{R}^{m\times p}$ with $p = 256$ elements, on a set of whitened normalized image patches of size $m = 8 \times 8$. Parameters are standard ones in this literature [72]—that is, a small value $\mu = 1e-5$, and $\lambda = 0.25$, leading to sparse matrices A (on average ≈ 4 non-zero coefficients per column of A). Note that our implementations are based on the open-source SPAMS toolbox [73].[2]

Neural networks. We consider now simple binary classification problems for learning neural networks. Assume that we are given a training set $\{a_i, b_i\}_{i=1}^n$, where the variables b_i in $\{-1, +1\}$ represent class labels, and a_i in $\mathbb{R}^p$ are feature vectors. The estimator of a label class is now given by a two-layer neural network $\hat{b} = \text{sign}(W_2^\top \sigma(W_1^\top a))$, where W_1 in $\mathbb{R}^{p\times d}$ represents the weights of a hidden layer with d neurons, W_2 in $\mathbb{R}^d$ carries the weight of the network's second layer, and $\sigma(u) = \log(1 + e^u)$ is a non-linear function, applied

[2] available here `http://spams-devel.gforge.inria.fr`.

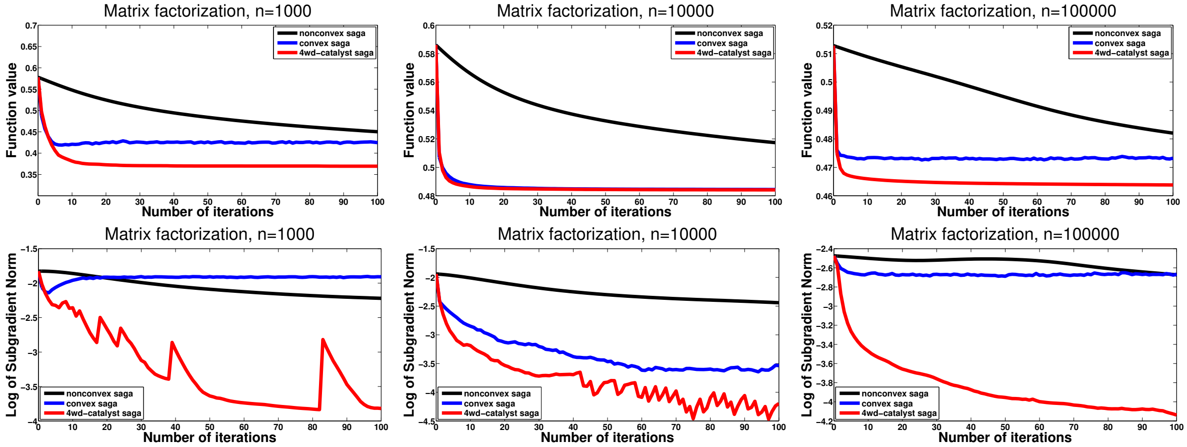


Figure 3.2: Dictionary learning experiments using SAGA. We plot the function value (top) and the subgradient norm (bottom). From left to right, we vary the size of dataset from $n = 1\,000$ to $n = 100\,000$.

pointwise to its arguments. We fix the number of hidden neurons to $d = 100$ and use the logistic loss to fit the estimators to the true labels. Since the memory required by SAGA becomes n times larger than SVRG for nonlinear models, which is problematic for large n, we can only perform experiments with SVRG. The experimental results are reported on two datasets alpha and covtype in Figures 3.3 and 3.4.

Initial estimates of L. The proposed algorithm 4WD-Catalyst-Automatic requires an initial estimate of the Lipschitz constant L. In the problems we are considering, there is no simple closed form formula available to compute an estimate of L. We use following heuristics to estimate L:

1. For matrix factorization, it can be shown that the function f_i defined in (3.22) is differentiable according to Danskin's theorem [see Bertsekas [5], Proposition B.25] and its gradient is given by

$$\nabla_D f_i(D) = -(x_i - D\alpha_i(D))\alpha_i(D)^T \quad \text{where} \quad \alpha_i(D) \in \operatorname*{argmin}_{\alpha \in \mathbb{R}^p} \frac{1}{2}\left\|x_i - D\alpha\right\|^2 + \psi(\alpha).$$

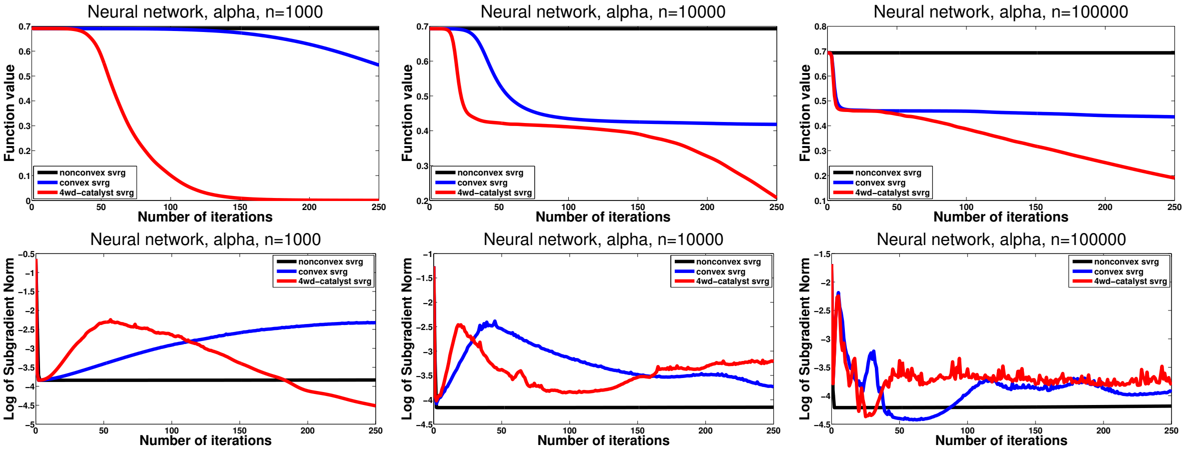


Figure 3.3: Neural network experiments on subsets of dataset alpha. From left to right, we vary the size of the dataset's subset from $n = 1\,000$ to $n = 100\,000$.

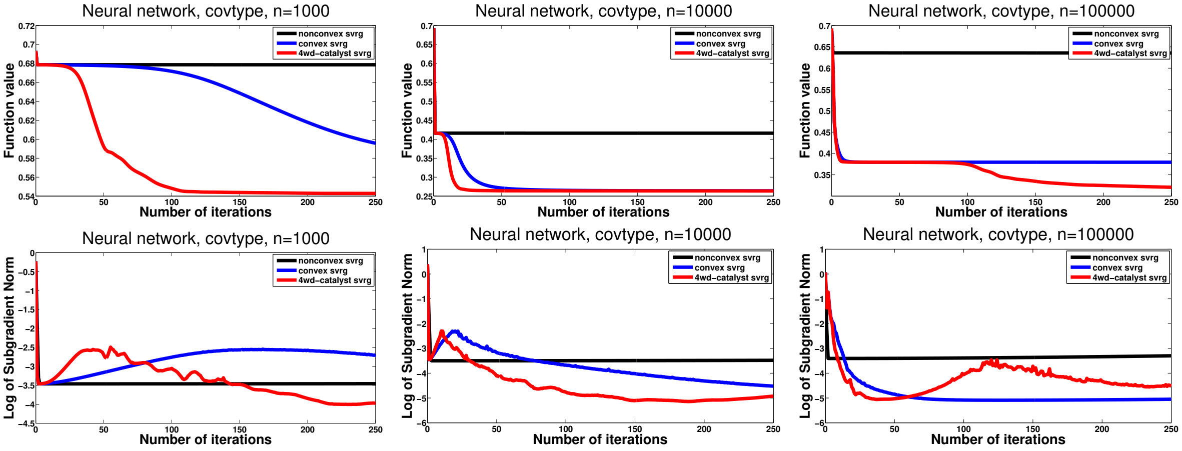


Figure 3.4: Neural network experiments on subsets of datasets alpha (top) and covtype (bottom).

If the coefficients α_i were fixed, the gradient would be linear in D and thus admit $\|\alpha_i\|^2$ as Lipschitz constant. Therefore, when initializing our algorithm at D_0, we find $\alpha_i(D_0)$ for any $i \in [1, n]$ and use $\max_{i\in[1,n]} \|\alpha_i(D_0)\|^2$ as an estimate of L.

2. For neural networks, the formulation we are considering is actually differentiable. We

randomly generates two pairs of weight vectors (W_1, W_2) and (W_1', W_2') and use the quantity

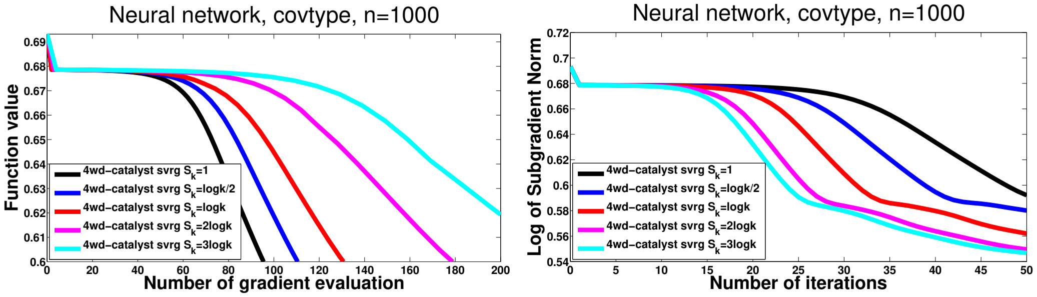
$$\max_{i \in [1,n]} \left\{ \frac{\|\nabla f_i(W_1, W_2) - \nabla f_i(W_1', W_2)\|}{\|W_1 - W_1'\|}, \frac{\|\nabla f_i(W_1, W_2) - \nabla f_i(W_1, W_2')\|}{\|W_2 - W_2'\|} \right\}$$

as an estimate of the Lipschitz constant, where f_i denotes the loss function respect to i-th training sample (a_i, b_i). We separate weights in each layer to estimate the Lipschitz constant *per layer*. Indeed the scales of the weights can be quite different across layers.

Computational cost. For the Convex-SVRG and Nonconvex-SVRG, one iteration corresponds to one pass over the data in the plots. On the one hand, since 4WD-Catalyst-Automatic-SVRG solves two sub-problems per iteration, the cost per iteration is twice that of the Convex-SVRG and Nonconvex-SVRG. On the other hand, in the experiments, we observe that, every time acceleration occurs, then $\tilde{x}_k$ is almost always preferred to $\bar{x}_k$ in step 4 of 4WD-Catalyst-Automatic, hence half of the computations are in fact not performed when running 4WD-Catalyst-Automatic-SVRG.

We report in Figure 3.5 an experimental study where we vary S on the neural network example. In terms of number of iterations, of course, the larger S_k the better the performance. This is not surprising as we solve each subproblem more accurately. Nevertheless, in terms of number of gradient evaluations, the relative performance is reversed. There is clearly no benefit to take larger S_k. This justifies in hindsight our choice of setting $S = 1$.

Experimental conclusions. In matrix factorization experiments, we observe that 4WD-Catalyst-Automatic-SVRG always outperforms the competing algorithms. Nonconvex-SVRG has slower convergence in objective values and Convex-SVRG is not always converging; see in particular right panel in Fig. 3.1. Therefore 4WD-Catalyst-Automatic-SVRG offers a more stable option than Convex-SVRG for minimizing nonconvex objectives. Furthermore, in these experiments 4WD-Catalyst-Automatic-SVRG enjoys a faster convergence in objective values. This confirms the remarkable ability of 4WD-Catalyst-Automatic-SVRG to adapt


Neural network, covtype, n=1000

Neural network, covtype, n=1000

Figure 3.5: We ran 50 iterations of 4WD-Catalyst-Automatic SVRG with different choices of S on a two-layer neural network. The data is a subset of dataset covtype. The x-axis is the number of gradient evaluations on the left, which is $T + S_k$ per iteration with $T = 1$; and the number of iterations on the right.

to nonconvex terrains. Similar conclusions hold when applying 4WD-Catalyst-Automatic to SAGA, which demonstrates how general 4WD-Catalyst-Automatic is.

In neural network experiments, we observe that 4WD-Catalyst-Automatic-SVRG converges much faster in terms of objective values than the competing algorithms. Nonconvex-SVRG with the theoretically recommended sequence of step-sizes [94] compares unfavorably here, which implies that the recommended step-sizes are too pessimistic hence too small. We also observe an interesting phenomenon: the subgradient norm may increase at some point then decrease, while the function value keeps decreasing, as the algorithm proceeds. This suggests that the extrapolation step, or the Auto-adapt procedure, is helpful to escape bad stationary points, *e.g.*, saddle-points. A more systematic study is required to confirm such observation, we leave it as a potential direction of future work.

Chapter 4

VARIATIONAL ANALYSIS OF SPECTRAL FUNCTIONS SIMPLIFIED

Joint work with D. Drusvyatskiy [40]

Abstract. Spectral functions of symmetric matrices – those depending on matrices only through their eigenvalues – appear often in optimization. A cornerstone variational analytic tool for studying such functions is a formula relating their subdifferentials to the subdifferentials of their diagonal restrictions. This paper presents a new, short, and revealing derivation of this result. We then round off the paper with an illuminating derivation of the second derivative of C^2-smooth spectral functions, highlighting the underlying geometry. All of our arguments have direct analogues for spectral functions of Hermitian matrices, and for singular value functions of rectangular matrices.

4.1 Introduction

This work revolves around *spectral functions.* These are functions on the space of $n \times n$ symmetric matrices $\mathbf{S}^n$ that depend on matrices only through their eigenvalues, that is, functions that are invariant under the action of the orthogonal group by conjugation. Spectral functions can always be written in a composite form $f \circ \lambda$, where f is a permutation-invariant function on $\mathbb{R}^n$ and λ is a mapping assigning to each matrix X the vector of eigenvalues $(\lambda_1(X), \ldots, \lambda_n(X))$ in nonincreasing order.

A pervasive theme in the study of such functions is that various variational properties of the permutation-invariant function f are inherited by the induced spectral function $f \circ \lambda$; see e.g. [28, 29, 31, 32, 37, 102, 106, 107]. Take convexity for example. Supposing that f is closed and convex, the main result of [59] shows that the Fenchel conjugate of $f \circ \lambda$ admits the elegant representation

$$(f \circ \lambda)^\star = f^\star \circ \lambda. \tag{4.1}$$

An immediate conclusion is that $f \circ \lambda$ agrees with its double conjugate and is therefore convex, that is, convexity of f is inherited by the spectral function $f \circ \lambda$. A convenient characterization of the subdifferential $\partial(f \circ \lambda)(X)$ in terms of $\partial f(\lambda(X))$ then readily follows [59, Theorem 3.1] — an important result for optimization specialists.

In a follow up paper [61], Lewis showed that even for nonconvex functions f, the following exact relationship holds:

$$\partial(f \circ \lambda)(X) = \{U(\operatorname{Diag} v)U^T : v \in \partial f(\lambda(X)),\, U \in \mathcal{O}^n_X\}, \tag{4.2}$$

where

$$\mathcal{O}^n_X := \{U \in \mathcal{O}^n : X = U(\operatorname{Diag} \lambda(X))U^T\}.$$

Here, the symbol $\mathcal{O}^n$ denotes the group of orthogonal matrices and the symbols $\partial(f \circ \lambda)$ and ∂f may refer to the Fréchet, limiting, or Clarke subdifferentials; see e.g. [98] for the relevant definitions. Thus calculating the subdifferential of the spectral function $f \circ \lambda$ on $\mathbf{S}^n$ reduces to computing the subdifferential of the usually much simpler function f on $\mathbb{R}^n$. For instance, subdifferential computation of the k'th largest eigenvalue function $X \mapsto \lambda_k(X)$ amounts to analyzing a piecewise polyhedral function, the k'th order statistic on $\mathbb{R}^n$ [61, Section 9]. Moreover, the subdifferential formula allows one to gauge the underlying geometry of spectral functions, through their "active manifolds" [28], for example.

In striking contrast to the convex case [59], the proof of the general subdifferential formula (4.2) requires much finer tools, and is less immediate to internalize. This paper presents a short, elementary, and revealing derivation of equation (4.2) that is no more involved than its convex counterpart. Here's the basic idea. Consider the *Moreau envelope*

$$f_\alpha(x) := \inf_y \big\{f(y) + \frac{1}{2\alpha}|x-y|^2\big\}.$$

Similar notation will be used for the envelope of $f \circ \lambda$. In direct analogy to equation (4.1), we will observe that the Moreau envelope satisfies the equation

$$(f \circ \lambda)_\alpha = f_\alpha \circ \lambda,$$

and derive a convenient formula for the corresponding proximal mapping. The case when f is an indicator function was treated in [29], and the argument presented here is a straightforward adaptation, depending solely on the Theobald–von Neumann inequality [108, 111]. The key observation now is independent of the eigenvalue setting: membership of a vector v in the proximal or in the Fréchet subdifferential of any function g at a point x is completely determined by the local behavior of the univariate function $\alpha \mapsto g_\alpha(x + \alpha v)$ near the origin. The proof of the subdifferential formula (4.2) quickly flows from there. It is interesting to note that the argument uses very little information about the properties of the eigenvalue map, with the exception of the Theobald–von Neumann inequality. Consequently, it applies equally well in a more general algebraic setting of certain isometric group actions, encompassing also an analogous subdifferential formula for functions of singular values derived in [64, 65, 101]; a discussion can be found in the appendix of the arXiv version of the paper. A different Lie theoretic approach in the convex case appears in [62].

We complete the paper by reconsidering the second-order theory of spectral functions. In [63, 102, 106], the authors derived a formula for the second derivative of a C^2-smooth spectral function. In its simplest form it reads

$$\nabla^2 F(\operatorname{Diag} a)[B] = \operatorname{Diag}\big(\nabla^2 f(a)\operatorname{diag}(B)\big) + \mathcal{A} \circ B,$$

where $\mathcal{A} \circ B$ is the Hadamard product and

$$\mathcal{A}_{ij} = \begin{cases} \frac{\nabla f(a)_i - \nabla f(a)_j}{a_i - a_j} & \text{if } a_i \neq a_j \\ \nabla^2 f(a)_{ii} - \nabla^2 f(a)_{ij} & \text{if } a_i = a_j \end{cases}.$$

This identity is quite mysterious, and its derivation is largely opaque geometrically. In the current work, we provide a transparent derivation, making clear the role of the invariance properties of the gradient graph. To this end, we borrow some ideas from [106], while giving them a geometric interpretation.

The outline of the manuscript is as follows. Section 4.2 records some basic notation and an important preliminary result about the Moreau envelope (Lemma 4.2.1). Section 4.3 contains background material on orthogonally invariant functions. Section 4.4 describes the derivation of the subdifferential formula and Section 4.5 focuses on the second-order theory – the main results of the paper.

4.2 Notation

This section briefly records some basic notation, following closely the monograph [98]. The symbol $\mathbb{E}$ will always denote an Euclidean space (finite-dimensional real inner product space) with inner product $\langle\cdot,\cdot\rangle$ and induced norm $|\cdot|$. A closed ball of radius $\varepsilon > 0$ around a point x will be denoted by $\mathcal{B}_\varepsilon(x)$. The closure and the convex hull of a set Q in $\mathbb{E}$ will be denoted by $\operatorname{cl} Q$ and $\operatorname{conv} Q$, respectively.

Throughout, we will consider functions f on $\mathbb{E}$ taking values in the extended real line $\overline{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$. For such a function f and a point $\bar{x}$, with $f(\bar{x})$ finite, the *proximal subdifferential* $\partial_p f(\bar{x})$ consists of all vectors $v \in \mathbb{E}$ such that there exists constants $r > 0$ and $\varepsilon > 0$ satisfying

$$f(x) \geq f(\bar{x}) + \langle v, x - \bar{x}\rangle - \frac{r}{2}|x - \bar{x}|^2 \qquad \text{for all } x \in \mathcal{B}_\varepsilon(\bar{x}).$$

Whenever f is C^2-smooth near $\bar{x}$, the proximal subdifferential $\partial_p f(\bar{x})$ consists only of the gradient $\nabla f(\bar{x})$. A function f is said to be *prox-bounded* if it majorizes some quadratic function. In particular, all lower-bounded functions are prox-bounded. For prox-bounded functions, the inequality in the definition of the proximal subdifferential can be taken to hold globally at the cost of increasing r [98, Proposition 8.46]. The *Fréchet subdifferential* of f at $\bar{x}$, denoted $\hat{\partial} f(\bar{x})$, consists of all vectors $v \in \mathbb{E}$ satisfying

$$f(x) \geq f(\bar{x}) + \langle v, x - \bar{x}\rangle + o(|x - \bar{x}|).$$

Here, as usual, $o(|x - \bar{x}|)$ denotes any term satisfying $\frac{o(|x-\bar{x}|)}{|x-\bar{x}|} \to 0$. Whenever f is C^1-smooth near $\bar{x}$, the set $\hat{\partial} f(\bar{x})$ consists only of the gradient $\nabla f(\bar{x})$. The subdifferentials $\partial_p f(\bar{x})$ and

$\hat{\partial} f(\bar{x})$ are always convex, while $\hat{\partial} f(\bar{x})$ is also closed. The *limiting subdifferential* of f at $\bar{x}$, denoted $\partial f(\bar{x})$, consists of all vectors $v \in \mathbb{E}$ so that there exist sequences x_i and $v_i \in \hat{\partial} f(x_i)$ with $(x_i, f(x_i), v_i) \to (\bar{x}, f(\bar{x}), v)$. The same object arises if the vectors v_i are restricted instead to lie in $\partial_p f(x_i)$ for each index i; see for example [98, Corollary 8.47]. The *horizon subdifferential*, denoted $\partial^\infty f(\bar{x})$, consists of all limits of $\lambda_i v_i$ for some sequences $v_i \in \partial f(x_i)$ and $\lambda_i \geq 0$ satisfying $x_i \to \bar{x}$ and $\lambda_i \searrow 0$. This object records horizontal "normals" to the epigraph of the function. For example, f is locally Lipschitz continuous around $\bar{x}$ if and only if the set $\partial^\infty f(\bar{x})$ contains only the zero vector.

The two key constructions at the heart of the paper are defined as follows. Given a function $f\colon \mathbb{E} \to \overline{\mathbb{R}}$ and a parameter $\alpha > 0$, the *Moreau envelope* f_α and the *proximal mapping* $P_\alpha f$ are defined by

$$\begin{aligned} f_\alpha(x) &:= \inf_{y \in \mathbb{E}} \left\{ f(y) + \frac{1}{2\alpha}|y - x|^2 \right\}, \\ P_\alpha f(x) &:= \operatorname*{argmin}_{y \in \mathbb{E}} \left\{ f(y) + \frac{1}{2\alpha}|y - x|^2 \right\}. \end{aligned}$$

Extending the definition slightly, we will set $f_0(x) := f(x)$. It is easy to see that f is *prox-bounded* if and only if there exists some point $x \in \mathbb{E}$ and a real $\alpha > 0$ satisfying $f_\alpha(x) > -\infty$.

The proximal and Fréchet subdifferentials are conveniently characterized by a differential property of the function $\alpha \mapsto f_\alpha(x + \alpha v)$. This observation is recorded below. To this end, for any function $\varphi\colon [0, \infty) \to \overline{\mathbb{R}}$, the one-sided derivative will be denoted by

$$\varphi'_+(0) := \lim_{\alpha \searrow 0} \frac{\varphi(\alpha) - \varphi(0)}{\alpha}.$$

Lemma 4.2.1 (Subdifferential and the Moreau envelope)**.**
Consider an lsc, prox-bounded function $f\colon \mathbb{E} \to \overline{\mathbb{R}}$, and a point x with $f(x)$ finite. Fix a vector $v \in \mathbb{E}$ and define the function $\varphi\colon [0, \infty) \to \overline{\mathbb{R}}$ by setting $\varphi(\alpha) := f_\alpha(x + \alpha v)$. Then the following are true.

(i) The vector v lies in $\hat{\partial} f(x)$ if and only if

$$\varphi'_+(0) = \frac{|v|^2}{2}. \tag{4.3}$$

(ii) The vector v lies in $\partial_p f(x)$ if and only if there exists $\alpha > 0$ satisfying $x \in P_\alpha f(x+\alpha v)$, or equivalently

$$\varphi(\alpha) = f(x) + \frac{|v|^2}{2}\alpha.$$

In this case, the equation above continues to hold for all $\tilde{\alpha} \in [0, \alpha]$.

Proof. Claim (ii) is immediate from definitions; see for example [98, Proposition 8.46]. Hence we focus on claim (i). To this end, note first that the inequality

$$\frac{f_\alpha(x+\alpha v) - f(x)}{\alpha} \leq \frac{|v|^2}{2} \qquad \text{holds for any } v \in \mathbb{E}. \tag{4.4}$$

Consider now a vector $v \in \hat{\partial} f(x)$ and any sequences $\alpha_i \searrow 0$ and $x_i \in P_{\alpha_i}(x + \alpha_i v)$. We may assume $x_i \neq x$ since otherwise there's nothing to prove. Clearly x_i tend to x and hence

$$\begin{aligned} f_{\alpha_i}(x+\alpha_i v) - f(x) &= f(x_i) - f(x) + \frac{1}{2\alpha_i}|(x_i - x) - \alpha_i v|^2 \\ &\geq o(|x_i - x|) + \frac{1}{2\alpha_i}|x_i - x|^2 + \frac{\alpha_i}{2}|v|^2. \end{aligned}$$

Consequently, we obtain the inequality

$$\frac{f_{\alpha_i}(x+\alpha_i v) - f(x)}{\alpha_i} \geq \frac{|x_i - x|}{\alpha_i} \cdot \frac{o(|x_i - x|)}{|x_i - x|} + \frac{1}{2}\left|\frac{x_i - x}{\alpha_i}\right|^2 + \frac{|v|^2}{2}.$$

Taking into account (4.4) yields the inequality

$$0 \geq \frac{|x_i - x|}{\alpha_i} \cdot \left(\frac{o(|x_i - x|)}{|x_i - x|} + \frac{1}{2}\left|\frac{x_i - x}{\alpha_i}\right|\right).$$

In particular, we deduce $\frac{x_i - x}{\alpha_i} \to 0$, and the equation (4.3) follows.

Conversely suppose that equation (4.3) holds, and for the sake of contradiction that v does not lie in $\hat{\partial} f(x)$. Then there exists $\kappa > 0$ and a sequence $y_i \to x$ satisfying

$$f(y_i) - f(x) - \langle v, y_i - x \rangle \leq -\kappa |y_i - x|.$$

Then for any $\alpha > 0$, observe

$$\begin{aligned} \frac{f_\alpha(x+\alpha v) - f(x)}{\alpha} &\leq \frac{1}{\alpha}\Big(f(y_i) - f(x) + \frac{1}{2\alpha}|(y_i - x) - \alpha v|^2\Big) \\ &\leq -\kappa \frac{|y_i - x|}{\alpha} + \frac{1}{2}\left|\frac{y_i - x}{\alpha}\right|^2 + \frac{|v|^2}{2}. \end{aligned}$$

Setting $\alpha_i := \frac{|y_i - x|}{\kappa}$ and letting i tend to ∞ yields a contradiction. □

4.3 Symmetry and orthogonal invariance

Next we recall a basic correspondence between symmetric functions and spectral functions of symmetric matrices. The discussion follows that of [61]. Henceforth $\mathbb{R}^n$ will denote an n-dimensional real Euclidean space with a specified basis. Hence one can associate $\mathbb{R}^n$ with a collection of n-tuples $(x_1, \ldots, x_n)$, in which case the inner product $\langle \cdot, \cdot \rangle$ is the usual dot product. The finite group of coordinate permutations of $\mathbb{R}^n$ will be denoted by Π^n. A function $f\colon \mathbb{R}^n \to \overline{\mathbb{R}}$ is *symmetric* whenever it is Π^n-invariant, meaning

$$f(\pi x) = f(x) \quad \text{for all } x \in \mathbb{R}^n \text{ and } \pi \in \Pi^n.$$

It is immediate to verify that if f is symmetric, then so is the Moreau envelope f_α for any $\alpha \geq 0$. This elementary observation will be important later.

The vector space of real $n \times n$ symmetric matrices will be denoted by $\mathbf{S}^n$ and will be endowed with the trace inner product $\langle X, Y \rangle = \operatorname{tr} XY$, and the induced Frobenius norm $|X| = \sqrt{\operatorname{tr} X^2}$. For any $x \in \mathbb{R}^n$, the symbol $\operatorname{Diag} x$ will denote the $n \times n$ matrix with x on its diagonal and with zeros off the diagonal, while for a matrix $X \in \mathbf{S}^n$, the symbol $\operatorname{diag} X$ will denote the n-vector of its diagonal entries.

The group of real $n \times n$ orthogonal matrices will be written as $\mathcal{O}^n$. The eigenvalue mapping $\lambda\colon \mathbf{S}^n \to \mathbb{R}^n$ assigns to each matrix X in $\mathbf{S}^n$ the vector of its eigenvalues $(\lambda_1(X), \ldots, \lambda_n(X))$ in a nonincreasing order. A function $F\colon \mathbf{S}^n \to \overline{\mathbb{R}}$ is *spectral* if it is $\mathcal{O}^n$-invariant under the conjugation action, meaning

$$F(UXU^T) = F(X) \quad \text{for all } X \in \mathbf{S}^n \text{ and } U \in \mathcal{O}^n.$$

In other words, spectral functions are those that depend on matrices only through their eigenvalues. A basic fact is that any spectral function F on $\mathbf{S}^n$ can be written as a composition of $F = f \circ \lambda$ for some symmetric function f on $\mathbb{R}^n$. Indeed, f can be realized as the restriction of F to diagonal matrices $f(x) = F(\operatorname{Diag} x)$.

Two matrices X and Y in $\mathbf{S}^n$ are said to admit a *simultaneous spectral decomposition* if there exists an orthogonal matrix $U \in \mathcal{O}^n$ such that UXU^T and UYU^T are both diagonal matrices. It is well-known that this condition holds if and only if X and Y commute. The matrices X and Y are said to admit a *simultaneous ordered spectral decomposition* if there exists an orthogonal matrix $U \in \mathcal{O}^n$ satisfying $UXU^T = \operatorname{Diag}\lambda(X)$ and $UYU^T = \operatorname{Diag}\lambda(Y)$. The following result characterizing this property, essentially due to Theobald [108] and von Neumann [111], plays a central role in spectral variation analysis.

Theorem 4.3.1 (Von Neumann-Theobald)**.** *Any two matrices X and Y in $\mathbf{S}^n$ satisfy the inequality*

$$|\lambda(X) - \lambda(Y)| \leq |X - Y|.$$

Equality holds if and only if X and Y admit a simultaneous ordered spectral decomposition.

This result is often called a trace inequality, since the eigenvalue mapping being 1-Lipschitz (as in the statement above) is equivalent to the inequality

$$\langle \lambda(X), \lambda(Y) \rangle \geq \langle X, Y \rangle \qquad \text{for all } X, Y \in \mathbf{S}^n.$$

4.4 Derivation of the subdifferential formula

In this section, we derive the subdifferential formula for spectral functions. In what follows, for any matrix $X \in \mathbf{S}^n$ define the diagonalizing matrix set

$$\mathcal{O}_X := \{U \in \mathcal{O}^n : U(\operatorname{Diag}\lambda(X))U^T = X\}.$$

The spectral subdifferential formula readily follows from Lemma 4.2.1 and the following intuitive proposition, a proof of which can essentially be seen in [29, Proposition 8].

Theorem 4.4.1 (Proximal analysis of spectral functions)**.**
Consider a symmetric function $f \colon \mathbb{R}^n \to \overline{\mathbb{R}}$. Then the equation

$$(f \circ \lambda)_\alpha = f_\alpha \circ \lambda \qquad \textit{holds.} \tag{4.5}$$

In addition, the proximal mapping admits the representation:

$$P_\alpha(f \circ \lambda)(X) = \big\{U\big(\mathit{Diag}\ y\big)U^T : y \in P_\alpha f(\lambda(X)),\, U \in \mathcal{O}_X\big\}. \tag{4.6}$$

Moreover, for any $Y \in P_\alpha(f \circ \lambda)(X)$ the matrices X and Y admit a simultaneous ordered spectral decomposition.

Proof. For any X and Y, applying the trace inequality (Theorem 4.3.1), we deduce

$$f(\lambda(Y)) + \frac{1}{2\alpha}|Y - X|^2 \geq f(\lambda(Y)) + \frac{1}{2\alpha}|\lambda(Y) - \lambda(X)|^2 \geq f_\alpha(\lambda(X)). \tag{4.7}$$

Taking the infimum over Y, we deduce $(f \circ \lambda)_\alpha(X) \geq f_\alpha(\lambda(X))$. On the other hand, for any $U \in \mathcal{O}_X$, the inequalities hold:

$$\begin{aligned}(f \circ \lambda)_\alpha(X) &= \inf_Y \left\{f(\lambda(Y)) + \frac{1}{2\alpha}|Y - X|^2\right\} \\ &= \inf_Y \left\{f(\lambda(Y)) + \frac{1}{2\alpha}|U^T Y U - \operatorname{Diag}\lambda(X)|^2\right\} \leq f_\alpha(\lambda(X)).\end{aligned}$$

This establishes (4.5).

To establish equation (4.6), consider first a matrix $U \in \mathcal{O}_X$ and a vector $y \in P_\alpha f(\lambda(X))$, and define $Y := U(\operatorname{Diag} y)U^T$. Then we have

$$(f \circ \lambda)(Y) + \frac{1}{2\alpha}|Y - X|^2 = f(y) + \frac{1}{2\alpha}|y - \lambda(X)|^2 = f_\alpha(\lambda(X)) = (f \circ \lambda)_\alpha(X).$$

Hence the inclusion $Y \in P_\alpha(f \circ \lambda)(X)$ is valid, as claimed. Conversely, fix any matrix $Y \in P_\alpha(f \circ \lambda)(X)$. Then plugging in Y into (4.7), the left-hand-side equals $(f \circ \lambda)_\alpha(X)$ and hence the two inequalities in (4.7) hold as equalities. The second equality immediately yields the inclusion $\lambda(Y) \in P_\alpha f(\lambda(X))$, while the first along with Theorem 4.3.1 implies that X and Y admit a simultaneous ordered spectral decomposition, as claimed. □

Combining Lemma 4.2.1 and Theorem 4.4.1, the main result of the paper readily follows.

Theorem 4.4.2 (Subdifferentials of spectral functions)**.** *Consider an lsc symmetric function $f\colon \mathbb{R}^n \to \overline{\mathbb{R}}$. Then the following equation holds:*

$$\partial(f \circ \lambda)(X) = \left\{U\big(\mathit{Diag}\, v\big)U^T : v \in \partial f(\lambda(X)),\, U \in \mathcal{O}_X\right\}. \tag{4.8}$$

Analogous formulas hold for the proximal, Fréchet, and horizon subdifferentials.

Proof. Fix a matrix X in the domain of $f \circ \lambda$ and define $x := \lambda(X)$. Without loss of generality, suppose that f is lower-bounded. Indeed if this were not the case, then since f is lsc there exists $\varepsilon > 0$ so that f is lower-bounded on the ball $\mathcal{B}_\varepsilon(x)$. Consequently adding to f the indicator function of the symmetric set $\cup_{\pi\in\Pi}\mathcal{B}_\varepsilon(\pi x)$ assures that the function is lower-bounded.

We first dispense with the easy inclusion $\subseteq$ for all the subdifferentials. To this end, recall that if V is a proximal subgradient of $f \circ \lambda$ at X, then there exists $\alpha > 0$ satisfying $X \in P_\alpha(f\circ\lambda)(X+\alpha V)$. Theorem 4.4.1 then implies that X and V commute. Taking limits, we deduce that all Fréchet, limiting, and horizon subgradients of $f \circ \lambda$ at X also commute with X. Recalling that commuting matrices admit simultaneous spectral decomposition, basic definitions immediately yield the inclusion $\subseteq$ in equation (4.8) for the proximal and for the Fréchet subdifferentials. Taking limits, we deduce the inclusion $\subseteq$ in (4.8) for the limiting and for the horizon subdifferentials, as well.

Next, we argue the reverse inclusion. To this end, define $V := U(\operatorname{Diag} v)U^T$ for an arbitrary matrix $U \in \mathcal{O}_X$ and any vector $v \in \mathbb{R}^n$. Then Theorem 4.4.1, along with the symmetry of the envelope f_α, yields the equation

$$\frac{(f\circ\lambda)_\alpha(X+\alpha V) - f(\lambda(X))}{\alpha} = \frac{f_\alpha(x+\alpha v) - f(x)}{\alpha}.$$

Consequently if v lies in $\partial_p f(x)$, then Lemma 4.2.1 shows that for some $\alpha > 0$ the right-hand-side equals $\frac{|v|^2}{2}$, or equivalently $\frac{|V|^2}{2}$. Lemma 4.2.1 then yields the inclusion $V \in \partial_p(f\circ\lambda)(X)$. Similarly if v lies in $\hat{\partial} f(x)$, then the same argument but with α tending to 0 shows that V lies in $\hat{\partial}(f\circ\lambda)(X)$. Thus the inclusion $\supseteq$ in equation (4.8) holds for the proximal and for the Fréchet subdifferentials. Taking limits, the same inclusion holds for the limiting and for the horizon subdifferentials. This completes the proof. □

Remark 8. *It easily follows from Theorem 4.4.2 that the inclusion $\supseteq$ holds for the Clarke subdifferential. The reverse inclusion, however, requires a separate argument given in [61, Sections 7-8].*

In conclusion, we should mention that all the arguments in the section apply equally well for Hermitian matrices (with the standard Hermitian trace product), with the orthogonal matrices replaced by unitary matrices. Entirely analogous arguments also apply for functions of singular values of rectangular matrices (real or complex). For more details, see the appendix in the arXiv version of the paper.

4.5 Hessians of C^2-smooth spectral functions

In this section, we revisit the second-order theory of spectral functions. To this end, fix for the entire section an lsc symmetric function $f\colon \mathbb{R}^n \to \overline{\mathbb{R}}$ and define the spectral function $F := f \circ \lambda$ on $\mathbf{S}^n$. It is well known that f is C^2-smooth around a matrix X if and only if F is C^2-smooth around $\lambda(X)$; see [63, 102, 106, 107]. Moreover, a formula for the Hessian of F is available: for matrices $A = \mathrm{Diag}(a)$ and $B \in \mathbf{S}^n$ we have

$$\nabla^2 F(A)[B] = \mathrm{Diag}\big(\nabla^2 f(a)\mathrm{diag}(B)\big) + \mathcal{A} \circ B,$$

where $\mathcal{A} \circ B$ is the Hadamard product and

$$\mathcal{A}_{ij} = \begin{cases} \frac{\nabla f(a)_i - \nabla f(a)_j}{a_i - a_j} & \text{if } a_i \neq a_j \\ \nabla^2 f(a)_{ii} - \nabla^2 f(a)_{ij} & \text{if } a_i = a_j \end{cases}.$$

The assumption that A is a diagonal matrix is made without loss of generality, as will be apparent shortly. In this section, we provide a transparent geometric derivation of the Hessian formula by considering invariance properties of $\mathrm{gph}\,\nabla F$. Some of our arguments give a geometric interpretation of the techniques in [106].

Remark 9 (Hessian and the gradient graph)**.** Throughout the section we will appeal to the following basic property of the Hessian. For any C^2-smooth function g on an Euclidean space, the vector $z := \nabla^2 g(a)[b]$ is the unique vector satisfying $(z, -b) \in N_{\mathrm{gph}\,\nabla g}(a, \nabla g(a))$.

Consider now the action of the orthogonal group $\mathcal{O}^n$ on $\mathbf{S}^n$ by conjugation namely $U.X = UXU^T$. Recall that F is *invariant* under this action, meaning $F(U.X) = F(X)$ for all orthogonal matrices U. This action naturally extends to the product space $\mathbf{S}^n \times \mathbf{S}^n$ by setting $U.(X,Y) = (U.X, U.Y)$. As we have seen, the graph $\operatorname{gph} \nabla F$ is then *invariant* with respect to this action:

$$U.\operatorname{gph} \nabla F = \operatorname{gph} \nabla F \qquad \text{for all } U \in \mathcal{O}^n.$$

One immediate observation is that $N_{\operatorname{gph} \nabla F}(U.X, U.Y) = U.N_{\operatorname{gph} \nabla F}(X,Y)$. Consequently we deduce

$$(Z, -B) \in N_{\operatorname{gph} \nabla F}(X,Y) \quad \Longleftrightarrow \quad (U.Z, -U.B) \in N_{\operatorname{gph} \nabla F}(U.X, U.Y)$$

The formula

$$\nabla^2 F(X)[B] = U^T.\nabla^2 F(U.X)[U.B] \tag{4.9}$$

now follows directly from Remark 9, whenever F is C^2-smooth around X. As a result, when speaking about the operator $\nabla^2 F(X)$, we may assume without loss of generality that X and $\nabla F(X)$ are both diagonal matrices.

Next we briefly recall a few rudimentary properties of the conjugation action; see for example [57, Sections 4, 8, 9]. We say that a $n \times n$ matrix W is *skew-symmetric* if $W^T = -W$. Then it is well-known that $\mathcal{O}^n$ is a smooth manifold and the tangent space to $\mathcal{O}^n$ at the identity matrix consists of skew-symmetric matrices:

$$T_{\mathcal{O}^n}(I) = \{W \in \mathbb{R}^{n \times n} : W \text{ is skew-symmetric}\}.$$

The *commutator* of two matrices $A, B \in \mathbb{R}^{n \times n}$, denoted by $[A,B]$ is the matrix $[A,B] := AB - BA$. An easy computation shows that the commutator of a skew-symmetric matrix with a symmetric matrix is itself symmetric. Moreover, the identity

$$\langle X, [W,Z] \rangle = \langle [X,W], Z \rangle$$

holds for any matrices $X, Z \in \mathbf{S}^n$ and skew-symmetric W. For any matrix $A \in \mathbf{S}^n$, the *orbit of A*, denoted by $\mathcal{O}^n.A$ is the set

$$\mathcal{O}^n.A = \{U.A \,:\, U \in \mathcal{O}^n\}.$$

Similarly, the orbit of a pair $(A, B) \in \mathbf{S}^n \times \mathbf{S}^n$ is the set

$$\mathcal{O}^n.(A, B) = \{(U.A, U.B) \,:\, U \in \mathcal{O}^n\}.$$

An standard computation[1] now shows that orbits are smooth manifolds with tangent spaces

$$T_{\mathcal{O}^n.A}(A) = \{[W, A] : W \text{ is skew-symmetric}\},$$
$$T_{\mathcal{O}^n.(A,B)}(A, B) = \{([W, A], [W, B]) : W \text{ is skew-symmetric}\}.$$

Now supposing that F is twice differentiable at a matrix $A \in \mathbf{S}^{n\times n}$, the graph $\operatorname{gph} \nabla F$ certainly contains the orbit $\mathcal{O}^n.(A, \nabla F(A))$. In particular, this implies that the tangent space to $\operatorname{gph} \nabla F$ at $(A, \nabla F(A))$ contains the tangent space to the orbit:

$$\{([W, A], [W, \nabla F(A)]) : W \text{ skew-symmetric}\}.$$

Consequently, for any $B \in \mathbf{S}^n$, the tuple $(\nabla^2 F(A)[B], -B)$ is orthogonal to the tuple $([W, A], [W, \nabla F(A)])$ for any skew-symmetric matrix W. We record this elementary observation in the following lemma. This also appears as [106, Lemma 3.2].

Lemma 4.5.1 (Orthogonality to orbits)**.** *Suppose F is C^2-smooth around $A \in \mathbf{S}^n$. Then for any skew-symmetric matrix W and any $B \in \mathbf{S}^n$, we have*

$$\langle \nabla^2 F(A)[B], [W, A]\rangle = \langle B, [W, \nabla F(A)]\rangle.$$

Proof. This is immediate from the preceding discussion. □

Next recall that the *stabilizer* of a matrix $A \in \mathbf{S}^n$ is the set:

$$\operatorname{Stab}(A) = \{U \in \mathcal{O}^n \,:\, U.A = A\}.$$

Similarly we may define the set Stab(A,B).

[1] Compute the differential of the mapping $\mathcal{O}^n \ni U \mapsto U.A$

Lemma 4.5.2 (Tangent space to the stabilizer)**.** *For any matrices $A, B \in \mathbf{S}^n$, the tangent spaces to Stab(A) and to Stab(A, B) at the identity matrix are the sets*

$$\{W \in \mathbb{R}^{n\times n} \, : \, W \text{ skew-symmetric, } [W, A] = 0\},$$

$$\Big\{W \in \mathbb{R}^{n\times n} \;\; : \, W \text{ skew-symmetric,} [W, A] = [W, B] = 0\Big\},$$

respectively.

(Proof sketch). Define the orbit map $\theta^{(A)} \colon \mathcal{O}^n \to \mathcal{O}^n.A$ by setting $\theta^{(A)}(U) := U.A$. A quick computation shows that $\theta^{(A)}$ is equivariant with respect to left-multiplication action of $\mathcal{O}^n$ on itself and the conjugation action of $\mathcal{O}^n$ on $\mathcal{O}^n.A$. Hence the equivariant rank theorem ([57, Theorem 7.25]) implies that $\theta^{(A)}$ has constant rank. In fact, since $\theta^{(A)}$ is surjective, it is a submersion. It follows that the stabilizer

$$\text{Stab}(A) = (\theta^{(A)})^{-1}(A)$$

is a smooth manifold with tangent space at the identity equal to the kernel of the differential $d\,\theta^{(A)}\big|_{U=I}(W) = [W, A]$. The expression for the tangent space to Stab(A) immediately follows. The analogous expression for Stab(A, B) follows along similar lines. □

With this, we are able to state and prove the main theorem.

Theorem 4.5.3 (Hessian of C^2-smooth spectral functions)**.** *Consider a symmetric function $f \colon \mathbb{R}^n \to \mathbb{R}$ and the spectral function $F = f \circ \lambda$. Suppose that F is C^2-smooth around a matrix $A :=$ Diag(a) and for any matrix matrix $B \in \mathbf{S}^n$ define $Z := \nabla^2 F(A)[B]$. Then equality*

$$\mathit{diag}(Z) = \nabla^2 f(a)[\mathit{diag}(B)],$$

holds, while for indices $i \neq j$, we have

$$Z_{ij} = \begin{cases} B_{ij}\left(\frac{\nabla f(a)_i - \nabla f(a)_j}{a_i - a_j}\right) & \text{if } a_i \neq a_j \\ B_{ij}\big(\nabla^2 f(a)_{ii} - \nabla^2 f(a)_{ij}\big) & \text{if } a_i = a_j. \end{cases}$$

Proof. First observe that clearly f must be C^2 smooth at a. Now, since A is diagonal, so is the gradient $\nabla F(A)$. So without loss of generality, we can assume $\nabla F(A) = \mathrm{Diag}(\nabla f(a))$.

Observe now that $(Z, -B)$ is orthogonal to the tangent space of $\mathrm{gph}\,\nabla F$ at $(A, \nabla F(A))$. On the other hand, for any vector $a' \in \mathbb{R}^n$, we have equality

$$\langle \begin{pmatrix} Z \\ \\ -B \end{pmatrix}, \begin{pmatrix} \mathrm{Diag}(a') - \mathrm{Diag}(a) \\ \\ \mathrm{Diag}(\nabla f(a')) - \mathrm{Diag}(\nabla f(a)) \end{pmatrix} \rangle = \langle \begin{pmatrix} \mathrm{diag}(Z) \\ \\ -\mathrm{diag}(B) \end{pmatrix}, \begin{pmatrix} a' - a \\ \\ \nabla f(a') - \nabla f(a) \end{pmatrix} \rangle.$$

It follows immediately that the tuple $(\mathrm{diag}(Z), -\mathrm{diag}(B))$ is orthogonal to the tangent space of $\mathrm{gph}\,\nabla f$ at $(a, \nabla f(a))$. Hence we deduce the equality $\mathrm{diag}(Z) = \nabla^2 f(a)[\mathrm{diag}(B)]$ as claimed.

Next fix indices i and j with $a_i \neq a_j$, and define the skew-symmetric matrix $W^{(i,j)} := e_i e_j^T - e_j e_i^T$, where e_k denotes the k'th standard basis vector. Applying Lemma 4.5.1 with the skew-symmetric matrix $W = \frac{1}{a_i - a_j} W^{(i,j)}$, we obtain

$$\begin{aligned} -2Z_{ij} &= \langle Z, \big[\tfrac{1}{a_i - a_j} W^{(i,j)}, A\big]\rangle = -\langle [\tfrac{1}{a_i - a_j} W^{i,j}, B\big], \nabla F(A)\rangle \\ &= -\langle \mathrm{diag}[\tfrac{1}{a_i - a_j} W^{i,j}, B\big], \nabla f(a)\rangle = -2B_{ij}\left(\frac{\nabla f(a)_i - \nabla f(a)_j}{a_i - a_j}\right). \end{aligned}$$

The claimed formula $Z_{ij} = B_{ij}\left(\frac{\nabla f(a)_i - \nabla f(a)_j}{a_i - a_j}\right)$ follows.

Finally, fix indices i and j, with $a_i = a_j$. Observe now the inclusion

$$\mathrm{Stab}(A) \subset \mathrm{Stab}(\nabla F(A)).$$

Indeed for any matrix $U \in \mathrm{Stab}(A)$, we have

$$\nabla F(A) = \nabla F(UAU^T) = U \nabla F(A) U^T.$$

This in particular immediately implies that the tangent space $T_{\mathrm{gph}\,\nabla F}(A, \nabla F(A))$ is invariant under the action of $\mathrm{Stab}(A)$, that is

$$U.T_{\mathrm{gph}\,\nabla F}(A, \nabla F(A)) = T_{\mathrm{gph}\,\nabla F}(A, \nabla F(A))$$

for any $U \in \mathrm{Stab}(A)$. Hence their entire orbit $\mathrm{Stab}(A).(X,Y)$ of any tangent vector $(X,Y) \in T_{\mathrm{gph}\,\nabla F}(A, \nabla F(A))$ is contained in the tangent space $T_{\mathrm{gph}\,\nabla F}(A, \nabla F(A))$. We conclude that the tangent space to such an orbit $\mathrm{Stab}(A).(X,Y)$ at (X,Y) is contained in $T_{\mathrm{gph}\,\nabla F}(A, \nabla F(A))$ as well.

Define now the matrices $E_i := \mathrm{Diag}(e_i)$ and $\hat{Z} := \mathrm{Diag}(\nabla^2 f(a)[e_i])$. Because F is C^2-smooth, clearly the inclusion $(E_i, \hat{Z}) \in T_{\mathrm{gph}\nabla F}(A, \nabla F(A)$ holds. The above argument, along with Lemma 4.5.2, immediately implies the inclusion

$$\{([W, E_i], [W, \hat{Z}]) : W \text{ skew-symmetric}, [W, A] = 0\} \quad \subseteq \quad T_{\mathrm{gph}\nabla F}(A, \nabla F(A))$$

and in particular, $([W, E_i], [W, \hat{Z}])$ is orthogonal to $(Z, -B)$ for any skew-symmetric W satisfying $[W, A] = 0$. To finish the proof, simply set $W = W^{(i,j)}$. Then since $a_i = a_j$, we have $[W, A] = 0$ and therefore

$$\begin{aligned} -2Z_{ij} = \langle Z, [W^{(i,j)}, E_i] \rangle = \langle B, [W^{(i,j)}, \hat{Z}] \rangle &= -\langle [W^{(i,j)}, B], \hat{Z} \rangle \\ &= -2B_{ij}\big(\nabla^2 f(a)_{ii} - \nabla^2 f(a)_{ij}\big), \end{aligned}$$

as claimed. This completes the proof. □

Remark 10. The appealing geometric techniques presented in this section seem promising for obtaining at least necessary conditions for the generalized Hessian, in the sense of [75], of spectral functions that are not necessarily C^2-smooth. Indeed the arguments presented deal entirely with the graph $\mathrm{gph}\,\nabla f$, a setting perfectly adapted to generalized Hessian computations. There are difficulties, however. To illustrate, consider a matrix $Z \in \partial^2 F(A|V)$. Then one can easily establish properties of $\mathrm{Diag}\, Z$ analogous to those presented in Theorem 4.5.3, as well as properties of Z_{ij} for indices i and j satisfying $a_i \neq a_j$. The difficulty occurs for indices i and j with $a_i = a_j$. In this case, our argument used explicitly the fact that tangent cones to $\mathrm{gph}\,\partial f$ are linear subspaces, a property that is decisively false in the general setting.

Appendix A

APPENDIX FOR CHAPTER 2

A.1 Proofs of Lemmas 2.5.3, 2.7.1 and Theorems 2.8.6, 2.8.7

In this section, we prove Lemmas 2.5.3, 2.7.1 and Theorems 2.8.6, 2.8.7 in order.

Proof of Lemma 2.5.3. Observe for any $t > 0$ and any proper, closed, convex function f, we have

$$\operatorname{prox}_{(tf)^\star}(w) = \operatorname*{argmin}_z \; \{tf^\star(z/t) + \tfrac{1}{2}\|z - w\|^2\} = t \cdot \operatorname{prox}_{f^\star/t}(w/t), \tag{A.1}$$

where the first equation follows from the definition of the proximal map and from [95, Theorem 16.1]. From [95, Theorem 31.5], we obtain $\operatorname{prox}_{th^\star}(w) = w - \operatorname{prox}_{(th^\star)^\star}(w)$, while an application of (A.1) with $f = h^\star$ then directly implies (2.29).

The fact that the gradient map $\nabla\Big(G^\star \circ A^* - \langle b, \cdot\rangle\Big)$ is Lipschitz with constant $t\|\nabla c(x)\|_{\text{op}}^2$ follows directly from $\nabla G^\star$ being t-Lipschitz continuous. The chain rule, in turn, yields

$$\nabla\Big(G^\star \circ A^* - \langle b, \cdot\rangle\Big)(w) = A\nabla G^\star(A^* w) - b.$$

Thus we must analyze the expression $\nabla G^\star(z) = \nabla(g + \frac{1}{2t}\|\cdot - x\|^2)^\star(z)$. Notice that the conjugate of $\frac{1}{2t}\|\cdot - x\|^2$ is the function $\frac{t}{2}\|\cdot\|^2 + \langle \cdot, x\rangle$. Hence, using [95, Theorem 16.4] we deduce

$$(g + \tfrac{1}{2t}\|\cdot - x\|^2)^\star(z) = \inf_y \; \{g^\star(y) + \tfrac{t}{2}\|z - y\|^2 + \langle z - y, x\rangle\} = (g^\star)_{1/t}(z + x/t) - \tfrac{1}{2t}\|x\|^2,$$

where the last equation follows from completing the square. We thus conclude

$$\nabla G^\star(z) = \nabla(g^\star)_{1/t}(z + x/t) = t \cdot \operatorname{prox}_{(g^\star/t)^\star}(z + x/t) = \operatorname{prox}_{tg}(x + tz),$$

where the second equality follows from Lemma 2.2.1 and the third from (A.1). The expressions (2.30) and (2.31) follow. □

Proof of Lemma 2.7.1. Observe

$$\|h(y) - h(z)\| \leq \frac{1}{m}\sum_{i=1}^{m} |h_i(y_i) - h_i(z_i)| \leq \frac{L}{m}\sum_{i=1}^{m} \|y - z\|_1 \leq \frac{L}{\sqrt{m}}\|y - z\|,$$

where the last equality follows from the l_p-norm comparison $\|\cdot\|_1 \leq \sqrt{m}\|\cdot\|_2$. This proves $\operatorname{lip}(h) \leq L/\sqrt{m}$. Next for any point x observe

$$\|\nabla c(x)\|_{\text{op}} = \max_{v:\|v\|=1} \|\nabla c(x)v\| \leq \sqrt{\sum_{i=1}^{m} \|\nabla c_i(x)\|^2} \leq \sqrt{m} \max_{i=1,\ldots,m} \|\nabla c_i(x)\|$$

By an analogous argument, we have

$$\|\nabla c(x) - \nabla c(z)\|_{\text{op}} \leq \sqrt{\sum_{i=1}^{m} \|\nabla c_i(x) - \nabla c_i(z)\|^2} \leq \beta\sqrt{m}\|x - z\|,$$

and hence $\operatorname{lip}(\nabla c) \leq \beta\sqrt{m}$. Finally, suppose that each h_i is C^1-smooth with L_h-Lipschitz gradient ∇h_i. Observe then

$$\|\nabla h(y) - \nabla h(z)\| = \frac{1}{m}\sqrt{\sum_{i=1}^{m} |h_i'(y_i) - h_i'(z_i)|^2} \leq \frac{L_h}{m}\|y - z\|^2.$$

The result follows. □

Proof of Theorem 2.8.6. The proof is a modification of the proof Theorem 2.8.3; as such, we skip some details. For any point w, we successively deduce

$$\begin{aligned}
F(x_k) &\leq h\big(\zeta_k + c(y_k) + \nabla c(y_k)(x_k - y_k)\big) + g(x_k) + \frac{\mu}{2}\|x_k - y_k\|^2 + L\varepsilon_k \\
&\leq \Big(h\big(\zeta_k + c(y_k) + \nabla c(y_k)(x_k - y_k)\big) + g(x_k) + \frac{\tilde{\mu}}{2}\|x_k - y_k\|^2\Big) \\
&\qquad - \frac{\tilde{\mu} - \mu}{2}\|x_k - y_k\|^2 + L\varepsilon_k \\
&\leq h\big(\zeta_k + c(y_k) + \nabla c(y_k)(w - y_k)\big) + g(w) \\
&\qquad + \frac{\tilde{\mu}}{2}\left(\|w - y_k\|^2 - \|w - x_k\|^2\right) - \frac{\tilde{\mu} - \mu}{2}\|x_k - y_k\|^2 + L\varepsilon_k \\
&\leq h\big(c(y_k) + \nabla c(y_k)(w - y_k)\big) + g(w) \\
&\qquad + \frac{\tilde{\mu}}{2}\left(\|w - y_k\|^2 - \|w - x_k\|^2\right) - \frac{\tilde{\mu} - \mu}{2}\|x_k - y_k\|^2 + 2L\varepsilon_k.
\end{aligned}$$

Setting $w := a_k v_k + (1-a_k)x_{k-1}$ and noting the equality $w - y_k = a_k(v_k - v_{k-1})$ then yields

$$\begin{aligned} F(x_k) \leq & h(c(y_k) + a_k \nabla c(y_k)(v_k - v_{k-1})) + a_k g(v_k) + (1-a_k)g(x_{k-1}) \\ & + \frac{\tilde{\mu}}{2}\left(\|a_k(v_k - v_{k-1})\|^2 - \|w - x_k\|^2\right) - \frac{\tilde{\mu}-\mu}{2}\|x_k - y_k\|^2 + 2L\varepsilon_k. \end{aligned}$$

Upper bounding $-\|w - x_k\|^2$ by zero and using Lipschitz continuity of h we obtain for any point x the inequalities

$$\begin{aligned} F(x_k) & \leq a_k\Big(\frac{1}{a_k}h(\xi_k + c(y_k) + a_k \nabla c(y_k)(v_k - v_{k-1})) + g(v_k)\Big) + (1-a_k)g(x_{k-1}) \\ & \quad + \frac{\tilde{\mu}a_k^2}{2}\|v_k - v_{k-1}\|^2 - \frac{\tilde{\mu}-\mu}{2}\|x_k - y_k\|^2 + L\delta_k + 2L\varepsilon_k. \\ & \leq a_k\Big(\frac{1}{a_k}h(\xi_k + c(y_k) + a_k \nabla c(y_k)(x - v_{k-1})) + g(x) \\ & \quad + \frac{\tilde{\mu}a_k}{2}(\|x - v_{k-1}\|^2 - \|v_k - v_{k-1}\|^2 - \|v_k - x\|^2)\Big) + (1-a_k)g(x_{k-1}) \\ & \quad + \frac{\tilde{\mu}a_k^2}{2}\|v_k - v_{k-1}\|^2 - \frac{\tilde{\mu}-\mu}{2}\|x_k - y_k\|^2 + L\delta_k + 2L\varepsilon_k. \\ & \leq h(c(y_k) + a_k \nabla c(y_k)(x - v_{k-1})) + a_k g(x) + \frac{\tilde{\mu}a_k^2}{2}(\|x - v_{k-1}\|^2 - \|v_k - x\|^2) \\ & \quad + (1-a_k)g(x_{k-1}) - \frac{\tilde{\mu}-\mu}{2}\|x_k - y_k\|^2 + 2L\delta_k + 2L\varepsilon_k. \end{aligned}$$

Define $\hat{x} := a_k x + (1-a_k)x_{k-1}$ and note $a_k(x - v_{k-1}) = \hat{x} - y_k$. The same argument as that of (2.77) yields

$$\begin{aligned} h(c(y_k) + \nabla c(y_k)(\hat{x} - y_k)) \leq & a_k h(c(x)) + (1-a_k)h(c(x_{k-1})) + \\ & \rho a_k(1-a_k)\|x - x_{k-1}\|^2 + \frac{r a_k^2}{2}\|x - v_{k-1}\|^2. \end{aligned}$$

Hence upper bounding $1 - a_k \leq 1$ we deduce

$$\begin{aligned} F(x_k) \leq & a_k F(x) + (1-a_k)F(x_{k-1}) + \frac{\tilde{\mu}a_k^2}{2}(\|x - v_{k-1}\|^2 - \|x - v_k\|^2) \\ & - \frac{\tilde{\mu}-\mu}{2}\|y_k - x_k\|^2 + \rho a_k\|x - x_{k-1}\|^2 + \frac{r a_k^2}{2}\|x - v_{k-1}\|^2 + 2L(\delta_k + \varepsilon_k). \end{aligned}$$

This expression is identical to that of (2.73) except for the error term $2L(\delta_k + \varepsilon_k)$. The same argument as in the proof of Theorem 2.8.3 then shows

$$\frac{F(x_N)-F(x^*)}{a_N^2}+\frac{\tilde{\mu}}{2}\|x^*-v_N\|^2 \leq \frac{\tilde{\mu}}{2}\left\|x^*-v_0\right\|^2+\rho M^2\left(\sum_{j=1}^N\frac{1}{a_j}\right) + \frac{NrM^2}{2}-\frac{\tilde{\mu}-\mu}{2}\sum_{j=1}^N\frac{\left\|x_j-y_j\right\|^2}{a_j^2}+2L\sum_{j=1}^N\frac{\varepsilon_j+\delta_j}{a_j^2}.$$

Hence appealing to Lemma 2.5.5, we deduce

$$\begin{aligned}\sum_{j=1}^N\frac{\|\mathcal{G}_{1/\tilde{\mu}}(y_j)\|^2}{a_j^2} &\leq 8L\tilde{\mu}\sum_{j=1}^N\frac{\varepsilon_j}{a_j^2}+2\sum_{j=1}^N\frac{\left\|\tilde{\mu}(x_j-y_j)\right\|^2}{a_j^2}\\ &\leq 8L\tilde{\mu}\sum_{j=1}^N\frac{\varepsilon_j}{a_j^2}+\frac{4\tilde{\mu}^2}{\tilde{\mu}-\mu}\left(\frac{\tilde{\mu}}{2}\|x^*-v_0\|^2+\frac{NM^2(r+\frac{\rho}{2}(N+3))}{2}+2L\sum_{j=1}^N\frac{\varepsilon_j+\delta_j}{a_j^2}\right).\end{aligned}$$

Therefore

$$\begin{aligned}\min_{i=1,\ldots,N}\|\mathcal{G}_{1/\tilde{\mu}}(y_j)\|^2 &\leq \frac{8\cdot 24L\tilde{\mu}\sum_{j=1}^N\frac{\varepsilon_j}{a_j^2}}{N(N+1)(2N+1)}\\ &+\frac{48\tilde{\mu}^2}{\tilde{\mu}-\mu}\left(\frac{\|x^*-v_0\|^2}{N(N+1)(2N+1)}+\frac{M^2(r+\frac{\rho}{2}(N+3))}{(N+1)(2N+1)}+\frac{4L\sum_{j=1}^N\frac{\varepsilon_j+\delta_j}{a_j^2}}{N(N+1)(2N+1)}\right)\end{aligned}$$

Combining the first and fourth terms and using the inequality $\tilde{\mu}\geq\mu$ yields the claimed efficiency estimate on $\|\mathcal{G}_{1/\tilde{\mu}}(y_j)\|^2$. Finally, the claimed efficiency estimate on the functional error $F(x_N)-F^*$ in the setting $r=0$ follows by the same reasoning as in Theorem 2.8.3. □

We next prove Theorem 2.8.7. To this end, we will need the following lemma.

Lemma A.1.1 (Lemma 1 in [100]). *Suppose the following recurrence relation is satisfied*

$$d_k^2 \leq d_0^2+c_k+\sum_{i=1}^k\beta_i d_i$$

for some sequences $d_i,\beta_i\geq 0$ and an increasing sequence $c_i\geq 0$. Then the inequality holds:

$$d_k\leq A_k:=\frac{1}{2}\sum_{i=1}^k\beta_i+\left(d_0^2+c_k+\left(\frac{1}{2}\sum_{i=1}^k\beta_i\right)^2\right)^{1/2}.$$

Moreover since the terms on the right-hand side increase in k, we also conclude for any $k\leq N$ the inequality $d_k\leq A_N$.

The *ε-subdifferential* of a function $f\colon \mathbb{R}^d \to \overline{\mathbb{R}}$ at a point $\bar{x}$ is the set

$$\partial_\varepsilon f(\bar{x}) := \{v \in \mathbb{R}^d : f(x) - f(\bar{x}) \geq \langle v, x - \bar{x}\rangle - \varepsilon \quad \text{for all } x \in \mathbb{R}^d\}.$$

In particular, notice that $\bar{x}$ is an ε-approximate minimizer of f if and only if the inclusion $0 \in \partial_\varepsilon f(\bar{x})$ holds. For the purpose of analysis, it is useful to decompose the function $F_{t,\alpha}(z, y, v)$ into a sum

$$F_{t,\alpha}(z; y, v) = F_\alpha(z; y, v) + \frac{1}{2t}\|z - v\|^2$$

The sum rule for ε-subdifferentials [53, Theorem 2.1] guarantees

$$\partial_\varepsilon F_{t,\alpha}(\cdot; y, v) \subseteq \partial_\varepsilon F_\alpha(\cdot; y, v) + \partial_\varepsilon \left(\frac{1}{2t}\| \cdot - v\|^2\right).$$

Lemma A.1.2. *The ε-subdifferential $\partial_\varepsilon \left(\frac{1}{2t}\| \cdot - v\|^2\right)$ at a point $\bar{z}$ is the set*

$$\left\{t^{-1}(z - v + \gamma) \,:\, \frac{1}{2t}\left\|\gamma\right\|^2 \leq \varepsilon\right\}.$$

Proof. This follows by completing the square in the definition of the ε-subdifferential. □

In particular, suppose that z^+ is an ε-approximate minimizer of $F_{t,\alpha}(\cdot; y, v)$. Then Lemma A.1.2 shows that there is a vector γ satisfying $\left\|\gamma\right\|^2 \leq 2t\varepsilon$ and

$$t^{-1}(v - z^+ - \gamma) \in \partial_\varepsilon F_\alpha(z^+; y, v). \tag{A.2}$$

We are now ready to prove Theorem 2.8.7.

Proof of Theorem 2.8.7. Let x_k, y_k, and v_k be the iterates generated by Algorithm 11. We imitate the proof of Theorem 2.8.3, while taking into account inexactness. First, inequality (2.74) is still valid:

$$F(x_k) \leq F(x_k; y_k) + \tfrac{\mu}{2}\left\|x_k - y_k\right\|^2.$$

Since x_k is an ε_k-approximate minimizer of the function $F(\cdot; y_k) = F_{1/\tilde{\mu},1}(\cdot; y_k, y_k)$, from (A.2), we obtain a vector γ_k satisfying $\left\|\gamma_k\right\|^2 \leq 2\varepsilon_k\tilde{\mu}^{-1}$ and $\tilde{\mu}(y_k - x_k - \gamma_k) \in \partial_{\varepsilon_k} F(x_k; y_k)$.

Consequently for all points w we deduce the inequality

$$F(x_k) \leq F(w; y_k) + \tfrac{\mu}{2}\left\|x_k - y_k\right\|^2 + \langle \tilde{\mu}(y_k - x_k - \gamma_k), x_k - w\rangle + \varepsilon_k. \tag{A.3}$$

Set $w_k := a_k v_k + (1 - a_k)x_{k-1}$ and define $c_k := x_k - w_k$. Taking into account $w_k - y_k = a_k(v_k - v_{k-1})$, the previous inequality with $w = w_k$ becomes

$$\begin{aligned} F(x_k) \leq{} & h(c(y_k) + a_k \nabla c(y_k)(v_k - v_{k-1})) + a_k g(v_k) + (1 - a_k)g(x_{k-1}) + \tfrac{\mu}{2}\left\|x_k - y_k\right\|^2 \\ & + \tilde{\mu}\langle y_k - x_k, c_k\rangle - \tilde{\mu}\langle \gamma_k, c_k\rangle + \varepsilon_k. \end{aligned} \tag{A.4}$$

By completing the square, one can check

$$\tilde{\mu}\langle y_k - x_k, c_k\rangle = \tfrac{\tilde{\mu}}{2}\big(\left\|a_k v_k - a_k v_{k-1}\right\|^2 - \left\|x_k - y_k\right\|^2 - \left\|c_k\right\|^2\big).$$

Observe in addition

$$-\tilde{\mu}\langle \gamma_k, c_k\rangle - \tfrac{\tilde{\mu}}{2}\left\|c_k\right\|^2 = -\tfrac{\tilde{\mu}}{2}\left\|\gamma_k + c_k\right\|^2 + \tfrac{\tilde{\mu}}{2}\left\|\gamma_k\right\|^2.$$

By combining the two equalities with (A.4) and dropping the term $\tfrac{\tilde{\mu}}{2}\left\|\gamma_k + c_k\right\|^2$, we deduce

$$\begin{aligned} F(x_k) \leq{} & h(c(y_k) + a_k \nabla c(y_k)(v_k - v_{k-1})) + a_k g(v_k) + (1 - a_k)g(x_{k-1}) \\ & + \tfrac{\tilde{\mu}a_k^2}{2}\left\|v_k - v_{k-1}\right\|^2 - \tfrac{\tilde{\mu}-\mu}{2}\left\|x_k - y_k\right\|^2 + \varepsilon_k + \tfrac{\tilde{\mu}}{2}\left\|\gamma_k\right\|^2. \end{aligned} \tag{A.5}$$

Next recall that v_k is a δ_k-approximate minimizer of $F_{(\tilde{\mu}a_k)^{-1}, a_k}(\cdot; y_k, v_{k-1})$. Using (A.2), we obtain a vector η_k satisfying $\left\|\eta_k\right\|^2 \leq \frac{2\delta_k}{a_k\tilde{\mu}}$ and $a_k\tilde{\mu}(v_{k-1} - v_k - \eta_k) \in \partial_{\delta_k} F_{a_k}(v_k; y_k, v_{k-1})$. Hence, we conclude for all the points x the inequality

$$\begin{aligned} F_{a_k}(v_k; y_k, v_{k-1}) \leq{} & \frac{1}{a_k} h(c(y_k) + a_k \nabla c(y_k)(x - v_{k-1}) + g(x) \\ & + \tilde{\mu}a_k\langle v_{k-1} - v_k - \eta_k, v_k - x\rangle + \delta_k. \end{aligned} \tag{A.6}$$

Completing the square, one can verify

$$\langle v_{k-1} - v_k, v_k - x\rangle = \frac{1}{2}(\left\|x - v_{k-1}\right\|^2 - \left\|x - v_k\right\|^2 - \left\|v_k - v_{k-1}\right\|^2).$$

Hence combining this with (A.5) and (A.6), while taking into account the inequalities $\|\gamma_k\|^2 \leq 2\varepsilon_k\tilde{\mu}^{-1}$ and $\|\eta_k\|^2 \leq \frac{2\delta_k}{a_k\tilde{\mu}}$, we deduce

$$\begin{aligned}F(x_k) \leq & h(c(y_k) + a_k\nabla c(y_k)(x - v_{k-1}) + a_k g(x) + (1-a_k)g(x_{k-1}) \\ & + \tfrac{\tilde{\mu}a_k^2}{2}(\|x - v_{k-1}\|^2 - \|x - v_k\|^2) + a_k\delta_k - \tfrac{\tilde{\mu}-\mu}{2}\|x_k - y_k\|^2 + 2\varepsilon_k \\ & + a_k^{3/2}\sqrt{2\tilde{\mu}\delta_k} \cdot \|v_k - x\|.\end{aligned}$$

Following an analogous part of the proof of Theorem 2.8.3, define now the point $\hat{x} = a_k x + (1-a_k)x_{k-1}$. Taking into account $a_k(x - v_{k-1}) = \hat{x} - y_k$, we conclude

$$\begin{aligned}h(c(y_k) + \nabla c(y_k)(\hat{x} - y_k)) &\leq (h \circ c)(\hat{x}) + \frac{r}{2}\|\hat{x} - y_k\|^2 \\ &\leq a_k h(c(x)) + (1-a_k)h(c(x_{k-1})) \\ &+ \rho a_k(1-a_k)\|x - x_{k-1}\|^2 + \frac{ra_k^2}{2}\|x - v_{k-1}\|^2.\end{aligned}$$

Thus we obtain

$$\begin{aligned}F(x_k) \leq & a_k F(x) + (1-a_k)F(x_{k-1}) + \rho a_k\|x - x_{k-1}\|^2 + \frac{ra_k^2}{2}\|x - v_{k-1}\|^2 \\ & + \tfrac{\tilde{\mu}a_k^2}{2}(\|x - v_{k-1}\|^2 - \|x - v_k\|^2) + a_k\delta_k - \tfrac{\tilde{\mu}-\mu}{2}\|x_k - y_k\|^2 + 2\varepsilon_k \\ & + a_k^{3/2}\sqrt{2\tilde{\mu}\delta_k} \cdot \|v_k - x\|.\end{aligned}$$

As in the proof of Theorem 2.8.3, setting $x = x^*$, we deduce

$$\begin{aligned}\frac{F(x_N) - F^*}{a_N^2} + \frac{\tilde{\mu}}{2}\|x^* - v_N\|^2 \leq & \frac{\tilde{\mu}}{2}\|x^* - v_0\|^2 + \rho M^2\sum_{i=1}^N \frac{1}{a_i} + \frac{NrM^2}{2} + \sum_{i=1}^N \frac{\delta_i}{a_i} \\ & - \frac{\tilde{\mu}-\mu}{2}\sum_{i=1}^N \frac{\|x_i - y_i\|^2}{a_i^2} + 2\sum_{i=1}^N \frac{\varepsilon_i}{a_i^2} + \sqrt{2\tilde{\mu}}\sum_{i=1}^N \|x^* - v_i\| \cdot \sqrt{\frac{\delta_i}{a_i}}.\end{aligned}$$

In particular, we have

$$\begin{aligned}\frac{\tilde{\mu}-\mu}{2}\sum_{i=1}^N \frac{\|x_i - y_i\|^2}{a_i^2} \leq & \frac{\tilde{\mu}}{2}\|x^* - v_0\|^2 + \frac{\rho M^2 N(N+3)}{4} + \frac{NrM^2}{2} + \sum_{i=1}^N \frac{\delta_i}{a_i} \\ & + 2\sum_{i=1}^N \frac{\varepsilon_i}{a_i^2} + \sqrt{2\tilde{\mu}}\sum_{i=1}^N \|x^* - v_i\| \cdot \sqrt{\frac{\delta_i}{a_i}}.\end{aligned} \tag{A.7}$$

and

$$\frac{\tilde{\mu}}{2}\|x^* - v_N\|^2 \leq \frac{\tilde{\mu}}{2}\|x^* - v_0\|^2 + \frac{\rho M^2 N(N+3)}{4} + \frac{NrM^2}{2} + \sum_{i=1}^{N} \frac{\delta_i}{a_i} + 2\sum_{i=1}^{N} \frac{\varepsilon_i}{a_i^2} + \sqrt{2\tilde{\mu}} \sum_{i=1}^{N} \|x^* - v_i\| \cdot \sqrt{\frac{\delta_i}{a_i}}.$$

Appealing to Lemma A.1.1 with $d_k = \|x^* - v_k\|$, we conclude $\|x^* - v_N\| \leq A_N$ for the constant

$$A_N := \sqrt{\frac{2}{\tilde{\mu}}} \sum_{i=1}^{N} \sqrt{\frac{\delta_i}{a_i}} + + \left(\|x^* - v_0\|^2 + \frac{M^2 N(r + \frac{\rho}{2}(N+3))}{\tilde{\mu}} + \frac{2}{\tilde{\mu}} \sum_{i=1}^{N} \frac{\delta_i}{a_i} + \frac{4}{\tilde{\mu}} \sum_{i=1}^{N} \frac{\varepsilon_i}{a_i^2} + \frac{2}{\mu} \left(\sum_{i=1}^{N} \sqrt{\frac{\delta_i}{a_i}} \right)^2 \right)^{1/2}.$$

Finally, combining inequality (A.7) with Lemma 2.5.1 we deduce

$$\frac{\tilde{\mu} - \mu}{2} \sum_{i=1}^{N} \frac{\|\mathcal{G}_{1/\tilde{\mu}}(y_i)\|^2}{a_i^2} \leq 2\tilde{\mu}(\tilde{\mu} - \mu) \sum_{i=1}^{N} \frac{\varepsilon_i}{a_i^2} + 2\tilde{\mu}^2 \Big(\frac{\tilde{\mu}}{2}\|x^* - v_0\|^2 + \frac{\rho M^2 N(N+3)}{4} + \frac{NrM^2}{2} + + \sum_{i=1}^{N} \frac{\delta_i}{a_i} + 2\sum_{i=1}^{N} \frac{\varepsilon_i}{a_i^2} + A_N \sqrt{2\tilde{\mu}} \sum_{i=1}^{N} \sqrt{\frac{\delta_i}{a_i}} \Big).$$

Hence

$$\min_{i=1,\ldots,N} \|\mathcal{G}_{1/\tilde{\mu}}(y_i)\|^2 \leq \frac{96\tilde{\mu} \sum_{i=1}^{N} \frac{\varepsilon_i}{a_i^2}}{N(N+1)(2N+1)} + \frac{96\tilde{\mu}^2}{\tilde{\mu} - \mu} \left(\frac{\tilde{\mu}\|x^* - v_0\|^2}{2N(N+1)(2N+1)} + \frac{M^2(r + \frac{\rho}{2}(N+3))}{2(N+1)(2N+1)} + \frac{\sum_{i=1}^{N} (\frac{\delta_i a_i + 2\varepsilon_i}{a_i^2}) + A_N \sqrt{2\tilde{\mu}} \sum_{i=1}^{N} \sqrt{\frac{\delta_i}{a_i}}}{N(N+1)(2N+1)} \right).$$

Combining the first and the fourth terms, the result follows. The efficiency estimate on $F(x_N) - F^*$ in the setting $r = 0$ follows by the same argument as in the proof of Theorem 2.8.3.

□

A.2 Backtracking

In this section, we present a variant of Algorithm 9 where the constants L and β are unknown. The scheme is recorded as Algorithm 16 and relies on a backtracking line-search, stated in Algorithm 15.

```
Algorithm 15: Backtracking($\eta, \alpha, t, y$)
  Initialize : A point $y$ and real numbers $\eta, \alpha \in (0,1)$ and $t > 0$.
  while $F(S_{\alpha t}(y)) > F_t(S_{\alpha t}(y))$ do
    $t \leftarrow \eta t$
  end
  Set $\tilde{\mu} = \frac{1}{\alpha t}$ and $x = S_{\alpha t}(y)$
  return $\tilde{\mu}, t, x$;
```

```
Algorithm 16: Accelerated prox-linear method with backtracking
  Initialize : Fix two points $x_0, v_0 \in \operatorname{dom} g$ and real numbers $t_0 > 0$ and $\eta, \alpha \in (0,1)$.
  Step k: $(k \geq 1)$ Compute
    $a_k = \frac{2}{k+1}$
    $y_k = a_k v_{k-1} + (1 - a_k) x_{k-1}$
    $(\tilde{\mu}_k, t_k, x_k) = \text{Backtracking}(\eta, \alpha, t_{k-1}, y_k)$
    $v_k = S_{\frac{1}{\tilde{\mu}_k a_k}, a_k}(y_k, v_{k-1})$
```

The backtracking procedure completes after only logarithmically many iterations.

Lemma A.2.1 (Termination of backtracking line search)**.** *Algorithm 15 on input* (η, α, t, y) *terminates after at most* $1 + \left\lceil \frac{\log(t\mu)}{\log(\eta^{-1})} \right\rceil$ *evaluations of* $S_{\alpha \cdot}(y)$.

Proof. This follows immediately by observing that the loop in Algorithm 15 terminates as soon as $t \leq \mu^{-1}$. □

We now establish convergence guarantees of Algorithm 16, akin to those of Algorithm 9.

Theorem A.2.2 (Convergence guarantees with backtracking)**.** *Fix real numbers* $t_0 > 0$ *and* $\eta, \alpha \in (0,1)$ *and let* x^* *be any point satisfying* $F(x^*) \leq F(x_k)$ *for all iterates* x_k *generated by Algorithm 16. Define* $\tilde{\mu}_{\max} := \max\{(\alpha t_0)^{-1}, (\alpha\eta)^{-1}\mu\}$ *and* $\tilde{\mu}_0 := (\alpha t_0)^{-1}$*. Then the efficiency estimate holds:*

$$\min_{j=1,\ldots,N} \left\|\mathcal{G}_{1/\tilde{\mu}_j}(y_j)\right\|^2 \leq \frac{24\tilde{\mu}_{\max}}{1-\alpha}\left(\frac{\tilde{\mu}_0\left\|x^* - v_0\right\|^2}{N(N+1)(2N+1)} + \frac{M^2\left(r + \frac{\rho}{2}(N+3)\right)}{(N+1)(2N+1)}\right).$$

In the case $r = 0$*, the inequality above holds with the second summand on the right-hand-side replaced by zero (even if* $M = \infty$*), and moreover the efficiency bound on function values holds:*

$$F(x_N) - F(x^*) \leq \frac{2\tilde{\mu}_{\max}\left\|x^* - v_0\right\|^2}{(N+1)^2}.$$

Proof. We closely follow the proofs of Lemma 2.8.5 and Theorem 2.8.3, as such, we omit some details. For $k \geq 1$, the stopping criteria of the backtracking algorithm guarantees that analogous inequalities (2.74) and (2.75) hold, namely,

$$F(x_k) \leq h\big(c(y_k) + \nabla c(y_k)(x_k - y_k)\big) + g(x_k) + \frac{1}{2t_k}\left\|x_k - y_k\right\|^2 \tag{A.8}$$

and

$$\begin{aligned} h\big(c(y_k) + \nabla c(y_k)(x_k - y_k)\big) + g(x_k) &\leq h\big(c(y_k) + \nabla c(y_k)(w_k - y_k)\big) \\ &\quad + \frac{\tilde{\mu}_k}{2}\left(\left\|w_k - y_k\right\|^2 - \left\|w_k - x_k\right\|^2 - \left\|x_k - y_k\right\|^2\right) \\ &\quad + a_k g(v_k) + (1-a_k) g(x_{k-1}) \end{aligned} \tag{A.9}$$

where $w_k := a_k v_k + (1-a_k)x_{k-1}$. By combining (A.8) and (A.9) together with the definition that $\tilde{\mu}_k = (\alpha t_k)^{-1}$, we conclude

$$\begin{aligned} F(x_k) &\leq h\big(c(y_k) + \nabla c(y_k)(w_k - y_k)\big) + a_k g(v_k) + (1-a_k) g(x_{k-1}) \\ &\quad + \frac{\tilde{\mu}_k}{2}\left(\left\|w_k - y_k\right\|^2 - \left\|w_k - x_k\right\|^2\right) + \frac{(1-\alpha^{-1})}{2t_k}\left\|x_k - y_k\right\|^2. \end{aligned} \tag{A.10}$$

We note the equality $w_k - y_k = a_k(v_k - v_{k-1})$. Observe that (2.76) holds by replacing $\frac{\tilde{\mu}}{2}$ with $\frac{\tilde{\mu}_k}{2}$; hence, we obtain for all points x

$$\begin{aligned} h\big(c(y_k) + a_k\nabla c(y_k)(v_k - v_{k-1})\big) + a_k g(v_k) &\leq h\big(c(y_k) + a_k \nabla c(y_k)(x - v_{k-1})\big) \\ &\quad + a_k g(x) + \frac{\tilde{\mu}_k a_k^2}{2}\left(\left\|x - v_{k-1}\right\|^2 - \left\|x - v_k\right\|^2 - \left\|v_k - v_{k-1}\right\|^2\right). \end{aligned} \tag{A.11}$$

Notice also that (2.77) holds as stated. Combining the inequalities (2.77), (A.10), and (A.11), we deduce

$$\begin{aligned}F(x_k) \leq & a_k F(x) + (1-a_k)F(x_{k-1}) + \frac{\tilde{\mu}_k a_k^2}{2}\left(\|x - v_{k-1}\|^2 - \|x - v_k\|^2\right) \\ & - \frac{(\alpha^{-1}-1)}{2t_k}\|y_k - x_k\|^2 + \rho a_k(1-a_k)\|x - x_{k-1}\|^2 + \frac{r a_k^2}{2}\|x - v_{k-1}\|^2.\end{aligned} \tag{A.12}$$

Plugging in $x = x^*$, subtracting $F(x^*)$ from both sides, and rearranging yields

$$\begin{aligned}\frac{F(x_k) - F(x^*)}{a_k^2} + \frac{\tilde{\mu}_k}{2}\|x^* - v_k\|^2 \leq & \frac{1-a_k}{a_k^2}(F(x_{k-1}) - F(x^*)) + \frac{\tilde{\mu}_k}{2}\|x^* - v_{k-1}\|^2 \\ & + \frac{\rho M^2}{a_k} + \frac{rM^2}{2} - \frac{(\alpha^{-1}-1)}{2t_k a_k^2}\|y_k - x_k\|^2.\end{aligned}$$

This is exactly inequality (2.78) with $\frac{\tilde{\mu}}{2}$ replaced by $\frac{\tilde{\mu}_k}{2}$ and $\frac{\tilde{\mu}-\mu}{2}$ replaced by $\frac{(\alpha^{-1}-1)}{2t_k}$; Using the fact that the sequence $\{\tilde{\mu}_k\}_{k=0}^{\infty}$ is nondecreasing and $\frac{1-a_k}{a_k^2} \leq \frac{1}{a_{k-1}^2}$, we deduce

$$\begin{aligned}\frac{F(x_k) - F(x^*)}{a_k^2} + \frac{\tilde{\mu}_k}{2}\|x^* - v_k\|^2 \leq \frac{\tilde{\mu}_k}{\tilde{\mu}_{k-1}} \Bigg(& \frac{F(x_{k-1}) - F(x^*)}{a_{k-1}^2} + \frac{\tilde{\mu}_{k-1}}{2}\|x^* - v_{k-1}\|^2 \\ & + \frac{\rho M^2}{a_k} + \frac{rM^2}{2} - \frac{(\alpha^{-1}-1)}{2t_k a_k^2}\|y_k - x_k\|^2 \Bigg).\end{aligned} \tag{A.13}$$

Notice $\tilde{\mu}_k \leq \alpha^{-1}\max\left\{t_0^{-1}, \eta^{-1}\mu\right\} =: \tilde{\mu}_{\max}$. Recursively applying (A.13) N times, we get

$$\begin{aligned}\frac{F(x_N) - F(x^*)}{a_N^2} + \frac{\tilde{\mu}_N}{2}\|x^* - v_N\|^2 \leq \left(\prod_{j=1}^{N}\frac{\tilde{\mu}_j}{\tilde{\mu}_{j-1}}\right)\Bigg(& \frac{\tilde{\mu}_0}{2}\|x^* - v_0\|^2 + \sum_{j=1}^{N}\frac{\rho M^2}{a_j} \\ & + \frac{NrM^2}{2} - \sum_{j=1}^{N}\frac{(\alpha^{-1}-1)}{2t_j}\cdot\frac{\|x_j - y_j\|^2}{a_j^2}\Bigg).\end{aligned} \tag{A.14}$$

By the telescoping property of $\prod_{j=1}^{N}\frac{\tilde{\mu}_j}{\tilde{\mu}_{j-1}} \leq \frac{\tilde{\mu}_{\max}}{\tilde{\mu}_0}$, we conclude

$$\begin{aligned}& \frac{\tilde{\mu}_{\max}}{\tilde{\mu}_0}\sum_{j=1}^{N}\frac{(\alpha^{-1}-1)}{2t_j}\cdot\frac{\|x_j - y_j\|^2}{a_j^2} \\ & \qquad \leq \frac{\tilde{\mu}_{\max}}{\tilde{\mu}_0}\left(\frac{\tilde{\mu}_0}{2}\|x^* - v_0\|^2 + \rho M^2\left(\sum_{j=1}^{N}\frac{1}{a_j}\right) + \frac{NrM^2}{2}\right).\end{aligned} \tag{A.15}$$

Using the inequality (A.15) and $\alpha t_j = \tilde{\mu}_j^{-1} \geq \tilde{\mu}_{\max}^{-1}$ for all j, we conclude

$$\left(\frac{(\alpha^{-1}-1)\alpha}{2\tilde{\mu}_0}\right)\left(\sum_{j=1}^{N}\frac{1}{a_j^2}\right)\min_{j=1,\ldots,N}\left\|\tilde{\mu}_j(x_j-y_j)\right\|^2$$
$$\leq \frac{\tilde{\mu}_{\max}}{\tilde{\mu}_0}\left(\frac{\tilde{\mu}_0}{2}\left\|x^*-v_0\right\|^2+\rho M^2\left(\sum_{j=1}^{N}\frac{1}{a_j}\right)+\frac{NrM^2}{2}\right).$$

The result follows by mimicking the rest of the proof in Theorem 2.8.3. Finally, suppose $r = 0$, and hence we can assume $\rho = 0$. Inequality (A.14) then implies

$$\frac{F(x_N)-F(x^*)}{a_N^2}+\frac{\tilde{\mu}_N}{2}\left\|x^*-v_N\right\|^2 \leq \frac{\tilde{\mu}_{\max}}{\tilde{\mu}_0}\cdot\frac{\tilde{\mu}_0}{2}\left\|x^*-v_0\right\|^2.$$

The claimed efficiency estimate follows. □

Appendix B
APPENDIX FOR CHAPTER 3

B.1 Convergence rates in strongly-convex composite minimization

We now briefly discuss convergence rates, which are typically given in different forms in the convex and non-convex cases. If the weak-convex constant is known, we can form a strongly convex approximation similar to [69]. For that purpose, we consider a strongly-convex composite minimization problem

$$\min_{x\in\mathbb{R}^p} \; h(x) := f_0(x) + \psi(x),$$

where $f_0\colon \mathbb{R}^p \to \mathbb{R}$ is μ-strongly convex and smooth with L-Lipschitz continuous gradient ∇f_0, and $\psi\colon \mathbb{R}^p \to \overline{\mathbb{R}}$ is a closed convex function with a computable proximal map

$$\mathrm{prox}_{\beta\psi}(y) := \operatorname*{argmin}_{z\in\mathbb{R}^p} \left\{ \psi(y) + \tfrac{1}{2\beta}\|z-y\|^2 \right\}.$$

Let x^* be the minimizer of h and h^* be the minimal value of h. In general, there are three types of measures of optimality that one can monitor: $\|x-x^*\|^2$, $h(x)-h^*$, and $\mathrm{dist}(0,\partial h(x))$.

Since h is strongly convex, the three of them are equivalent in terms of convergence rates if one can take an extra *prox-gradient step*:

$$[x]_L := \mathrm{prox}_{\psi/L}(x - L^{-1}\nabla f_0(x)).$$

To see this, define the *displacement vector*, also known as the gradient mapping, $g_L(x) := L(x-[x]_L)$, and notice the inclusion $g_L(x) \in \partial h([x]_L)$. In particular $g_L(x) = 0$ if and only if x is the minimizer of h. These next inequalities follow directly from Theorem 2.2.7 in [81]:

$$\begin{aligned}
\tfrac{1}{2L}\|g_L(x)\| \leq& \|x-x^*\| \leq \tfrac{2}{\mu}\|g_L(x)\| \\
\tfrac{\mu}{2}\|x-x^*\|^2 \leq& h(x)-h^* \leq \tfrac{1}{2\mu}|\partial h(x)|^2 \\
2\mu(h([x]_L)-h^*) \leq& \|g_L(x)\|^2 \leq 2L(h(x)-h([x]_L))
\end{aligned}$$

Thus, an estimate of any one of the four quantities $\|x - x^*\|$, $h(x) - h^*$, $\|g_L(x)\|$, or $\text{dist}(0, \partial h(x))$ directly implies an estimate of the other three evaluated either at x or at $[x]_L$.

B.2 Theoretical analysis of the basic algorithm

We present here proofs of the theoretical results of the paper. Althroughout the proofs, we shall work under the Assumptions on f stated in Section 3.3 and the Assumptions on $\mathcal{M}$ stated in Section 3.4.

B.2.1 Convergence guarantee of 4WD-Catalyst

In Theorem 3.3.1 and Theorem 3.3.2 under an appropriate tolerance policy on the proximal subproblems (3.5) and (3.7), 4WD-Catalyst performs no worse than an exact proximal point method in general, while automatically accelerating when f is convex. For this, we need the following observations.

Lemma B.2.1 (Growth of (α_k))**.** *Suppose the sequence $\{\alpha_k\}_{k\geq 1}$ is produced by Algorithm 12. Then, the following bounds hold for all $k \geq 1$:*

$$\frac{\sqrt{2}}{k+2} \leq \alpha_k \leq \frac{2}{k+1}.$$

Proof. This result is noted without proof in a remark of [109]. For completeness, we give below a simple proof using induction. Clearly, the statement holds for $k = 1$. Assume the inequality on the right-hand side holds for k. By using the induction hypothesis, we get

$$\alpha_{k+1} = \frac{\sqrt{\alpha_k^4 + 4\alpha_k^2} - \alpha_k^2}{2} = \frac{2}{\sqrt{1 + 4/\alpha_k^2} + 1} \leq \frac{2}{\sqrt{1 + (k+1)^2} + 1} \leq \frac{2}{k+2},$$

as claimed and the expression for α_{k+1} is given by explicitly solving (3.9). To show the lower bound, we note that for all $k \geq 1$, we have

$$\alpha_{k+1}^2 = (1 - \alpha_{k+1})\alpha_k^2 = \prod_{i=2}^{k+1}(1 - \alpha_i)\alpha_1^2 = \prod_{i=2}^{k+1}(1 - \alpha_i).$$

Using the established upper bound $\alpha_k \leq \frac{2}{k+1}$ yields

$$\alpha_{k+1}^2 \geq \prod_{i=2}^{k+1} \left(1 - \frac{2}{i+1}\right) = \frac{2}{(k+2)(k+1)} \geq \frac{2}{(k+2)^2}.$$

The result follows. $\square$

Lemma B.2.2 (Prox-gradient and near-stationarity)**.** *If y^+ satisfies dist$(0, \partial f_\kappa(y^+; y)) < \varepsilon$, then the following inequality holds:*

$$\operatorname{dist}\big(0, \partial f(y^+)\big) \leq \varepsilon + \left\|\kappa(y^+ - y)\right\|.$$

Proof. We can find $\xi \in \partial f_\kappa(y^+; y)$ with $\|\xi\| \leq \varepsilon$. Taking into account $\partial f_\kappa(y^+; y) = \partial f(y^+) + \kappa(y^+ - y)$ the result follows. $\square$

Next we establish convergence guarantees of Theorem 3.3.1 and Theorem 3.3.2 for 4WD-Catalyst .

Proof of Theorem 3.3.2 and Theorem 3.3.2. The proof of Theorem 3.3.1 follows the analysis of inexact proximal point method [69, 51, 6]. The descent condition in (3.11) implies $\{f(x_k)\}_{k\geq 0}$ are monotonically decreasing. From this, we deduce

$$f(x_{k-1}) = f_\kappa(x_{k-1}; x_{k-1}) \geq f_\kappa(\bar{x}_k; x_{k-1}) \geq f(x_k) + \frac{\kappa}{2} \left\|\bar{x}_k - x_{k-1}\right\|^2. \tag{B.1}$$

Using the adaptive stationarity condition (3.11), we apply Lemma B.2.2 with $y = x_{k-1}$, $y^+ = \bar{x}_k$ and $\varepsilon = \kappa \left\|\bar{x}_k - x_{k-1}\right\|$; hence we obtain

$$\operatorname{dist}(0, \partial f(\bar{x}_k)) \leq 2 \left\|\kappa(\bar{x}_k - x_{k-1})\right\|.$$

We combine the above inequality with (B.1) to deduce

$$\operatorname{dist}^2(0, \partial f(\bar{x}_k)) \leq 4 \left\|\kappa(\bar{x}_k - x_{k-1})\right\|^2 \leq 8\kappa \left(f(x_{k-1}) - f(x_k)\right). \tag{B.2}$$

Summing $j = 1$ to N, we conclude

$$\begin{aligned}\min_{j=1,\ldots,N} \left\{\operatorname{dist}^2(0, \partial f(\bar{x}_j))\right\} &\leq \frac{4}{N}\sum_{j=1}^{N} \|\kappa(\bar{x}_k - x_{k-1})\|^2) \\ &\leq \frac{8\kappa}{N}\left(\sum_{j=1}^{N} f(x_{j-1}) - f(x_j)\right) \\ &\leq \frac{8\kappa}{N}\left(f(x_0) - f^*\right).\end{aligned}$$

Next, suppose the function f is convex. Our analysis is similar to that of [109, 4]. Using the stopping criteria (3.12), fix an $\xi_k \in \partial f_\kappa(\tilde{x}_k; y_k)$ with $\|\xi_k\| < \frac{\kappa}{k+1}\|\tilde{x}_k - y_k\|$. For any $x \in \mathbb{R}^n$, Equation (3.10), and the strong convexity of the function $f_\kappa(\cdot; y_k)$ yields

$$f(x_k) \leq f(\tilde{x}_k) \leq f(x) + \frac{\kappa}{2}\left(\|x - y_k\|^2 - \|x - \tilde{x}_k\|^2 - \|\tilde{x}_k - y_k\|^2\right) + \xi_k^T\left(\tilde{x}_k - x\right).$$

We substitute $x = \alpha_k x^* + (1 - \alpha_k)x_{k-1}$ where x^* is any minimizer of f. Using the convexity of f, the norm of ξ_k, and Equations (3.6) and (3.8), we deduce

$$\begin{aligned}f(x_k) \leq \alpha_k f(x^*) + (1 - \alpha_k)f(x_{k-1}) &+ \frac{\alpha_k^2\kappa}{2}\left(\|x^* - v_{k-1}\|^2 - \|x^* - v_k\|^2\right) \\ &- \frac{\kappa}{2}\|\tilde{x}_k - y_k\|^2 + \frac{\alpha_k\kappa}{k+1}\|\tilde{x}_k - y_k\|\,\|x^* - v_k\|.\end{aligned} \tag{B.3}$$

Set $\theta_k = \frac{1}{k+1}$. Completing the square on Equation (B.3), we obtain

$$\frac{-\kappa}{2}\|\tilde{x}_k - y_k\|^2 + \alpha_k\theta_k\kappa\|\tilde{x}_k - y_k\|\,\|x^* - v_k\| \leq \frac{\kappa}{2}\left(\alpha_k\theta_k\right)^2\|x^* - v_k\|^2.$$

Hence, we deduce

$$\begin{aligned}f(x_k) - f^* \leq (1 - \alpha_k)(f(x_{k-1}) - f^*) &+ \frac{\alpha_k^2\kappa}{2}\left(\|x^* - v_{k-1}\|^2 - \|x^* - v_k\|^2\right) \\ &+ \frac{\kappa}{2}\left(\alpha_k\theta_k\right)^2\|x^* - v_k\|^2. \\ = (1 - \alpha_k)(f(x_{k-1}) - f^*) &+ \frac{\alpha_k^2\kappa}{2}\left(\|x^* - v_{k-1}\|^2 - \left(1 - \theta_k^2\right)\|x^* - v_k\|^2\right)\end{aligned}$$

Denote $A_k := 1 - \theta_k^2$. Subtracting f^* from both sides and using the inequality $\frac{1-\alpha_k}{\alpha_k^2} = \frac{1}{\alpha_{k-1}^2}$

and $\alpha_1 \equiv 1$, we derive the following recursion argument:

$$\begin{aligned}\frac{f(x_k) - f^*}{\alpha_k^2} + \frac{A_k \kappa}{2} \|x^* - v_k\|^2 &\leq \frac{1-\alpha_k}{\alpha_k^2}\big(f(x_{k-1}) - f^*\big) + \frac{\kappa}{2}\|x^* - v_{k-1}\|^2 \\ &\leq \frac{1}{A_{k-1}}\left(\frac{f(x_{k-1}) - f^*}{\alpha_{k-1}^2} + \frac{A_{k-1}\kappa}{2}\|x^* - v_{k-1}\|^2\right).\end{aligned}$$

The last inequality follows because $0 < A_{k-1} \leq 1$. Iterating N times,we deduce

$$\frac{f(x_N) - f^*}{\alpha_N^2} \leq \prod_{j=2}^{N} \frac{1}{A_{j-1}} \left(\frac{\kappa}{2}\|x^* - v_0\|^2\right). \tag{B.4}$$

We note

$$\prod_{j=2}^{N} \frac{1}{A_{j-1}} = \frac{1}{\prod_{j=2}^{N}\left(1 - \frac{1}{(j+1)^2}\right)} \leq 2; \tag{B.5}$$

thereby concluding the result. Summing up (B.2) from $j = N+1$ to $2N$, we obtain

$$\begin{aligned}\min_{j=1,\ldots,2N} \left\{\text{dist}^2(0, \partial f(\bar{x}_j))\right\} &\leq \frac{4}{N}\sum_{j=N+1}^{2N} \|\kappa(\bar{x}_k - x_{k-1})\|^2) \\ &\leq \frac{8\kappa}{N}\left(\sum_{j=N+1}^{2N} f(x_{j-1}) - f(x_j)\right) \\ &\leq \frac{8\kappa}{N}\left(f(x_N) - f^*\right)\end{aligned}$$

Combining this inequality with (B.4), the result is shown.

□

B.3 Analysis of 4WD-Catalyst-Automatic and **Auto-adapt**

Linear convergence interlude. Our assumption on the linear rate of convergence of $\mathcal{M}$ (see (3.14)) may look strange at first sight. Nevertheless, most linearly convergent first-order methods $\mathcal{M}$ for composite minimization either already satisfy this assumption or can be made to satisfy it by introducing an extra prox-gradient step. To see this, recall the convex composite minimization problem from Section B.1

$$\min_{z \in \mathbb{R}^p} h(z) := f_0(z) + \psi(z),$$

where

1. $f_0 \colon \mathbb{R}^p \to \mathbb{R}$ is convex and C^1-smooth with the gradient ∇f_0 that is L-Lipschitz,

2. $\psi \colon \mathbb{R}^p \to \overline{\mathbb{R}}$ is a closed convex function with a computable proximal map

$$\mathrm{prox}_{\beta\psi}(y) := \operatorname*{argmin}_{z} \; \{\psi(y) + \tfrac{1}{2\beta}\|z - y\|^2\}.$$

See [89] for a survey of proximal maps. Typical linear convergence guarantees of an optimization algorithm assert existence of constants $A \in \mathbb{R}$ and $\tau \in (0,1)$ satisfying

$$h(z_t) - h^* \leq A(1-\tau)^t(h(z_0) - h^*) \tag{B.6}$$

for each $t = 0, 1, 2, \ldots, \infty$. To bring such convergence guarantees into the desired form (3.14), define the prox-gradient step

$$[z]_L := \mathrm{prox}_{\psi/L}(z - L^{-1}\nabla f_0(z)),$$

and the displacement vector

$$g_L(z) = L(z - [z]_L),$$

and notice the inclusion $g_L(z) \in \partial h([z]_L)$. The following inequality follows from [86]:

$$\|g_L(z)\|^2 \leq 2L(h(z) - h([z]_L)) \leq 2L(h(z) - h^*).$$

Thus, the linear rate of convergence (B.6) implies

$$\|g_L(z_t)\|^2 \leq 2LA(1-\tau)^t(h(z_0) - h^*),$$

which is exactly in the desired form (3.14).

B.3.1 Convergence analysis of the adaptive algorithm: 4WD-Catalyst-Automatic

First, under some reasonable assumptions on the method $\mathcal{M}$ (see Section 3.4.1), the sub-method Auto-adapt terminates.

Lemma B.3.1 (Auto-adapt terminates)**.** *Assume that $\tau_\kappa \to 1$ when $\kappa \to +\infty$. The procedure Auto-adapt$(x, \kappa, \varepsilon, T)$ terminates after finitely many iterations.*

Proof. Due to our assumptions on $\mathcal{M}$ and the expressions $f_\kappa(x; x) = f(x)$ and $f_\kappa^*(x) \geq f^*$, we have

$$\operatorname{dist}^2\big(0, \partial f_\kappa(z_T; x)\big) \leq A(1-\tau_\kappa)^T \big(f(x) - f_\kappa^*(x)\big) \leq A(1-\tau_\kappa)^T \big(f(x) - f^*\big). \tag{B.7}$$

Since τ_κ tends to one, for all sufficiency large κ, we can be sure that the right-hand-side is smaller than ε^2. On the other hand, for $\kappa > \rho$, the function $f_\kappa(\cdot; x)$ is $(\kappa - \rho)$-strongly convex and therefore we have $\operatorname{dist}^2(0, \partial f_\kappa(z_T; x)) \geq 2(\kappa - \rho)(f_\kappa(z_T; x) - f_\kappa^*(x))$. Combining this with (B.7), we deduce

$$f_\kappa(z_T; x) - f_\kappa^*(x) \leq \frac{A(1-\tau_\kappa)^T}{2(\kappa - \rho)} \big(f(x) - f_\kappa^*(x)\big).$$

Letting $\kappa \to \infty$, we deduce $f_\kappa(z_T; x) \leq f(x)$, as required. Thus the loop indeed terminates. □

We prove the main result, Theorem 3.4.3, for 4WD-Catalyst-Automatic.

Proof of Theorem 3.4.3. The proof closely resembles the proofs of Theorem 3.3.2 and Theorem 3.3.2, so we omit some of the details. The main difference in the proof is that we keep track of the effects the parameters κ_{cvx} and κ_0 have on the inequalities as well as the sequence of κ_k. Since $\{f(x_k)\}_{k \geq 0}$ are monotonically decreasing, we deduce

$$f(x_{k-1}) = f_{\kappa_k}(x_{k-1}; x_{k-1}) \geq f_{\kappa_k}(\bar{x}_k; x_{k-1}) \geq f(x_k) + \frac{\kappa_k}{2} \left\| \bar{x}_k - x_{k-1} \right\|^2. \tag{B.8}$$

Using the adaptive stationary condition (3.18), we take $\varepsilon = \kappa_k \left\| \bar{x}_k - x_{k-1} \right\|$ in Lemma B.2.2 to obtain

$$\operatorname{dist}(0, \partial f(\bar{x}_k)) \leq 2 \left\| \kappa_k (\bar{x}_k - x_{k-1}) \right\|.$$

We combine the above inequality with (B.8) to deduce

$$\operatorname{dist}^2(0, \partial f(\bar{x}_k)) \leq 4 \left\| \kappa_k (\bar{x}_k - x_{k-1}) \right\|^2 \leq 8\kappa_{\max} \left(f(x_{k-1}) - f(x_k) \right). \tag{B.9}$$

Summing $j = 1$ to N, we conclude

$$\begin{aligned}
\min_{j=1,\ldots,N} \left\{\text{dist}^2(0, \partial f(\bar{x}_j))\right\} &\leq \frac{4}{N}\sum_{j=1}^{N} 2\left\|\kappa_k(\bar{x}_k - x_{k-1})\right\|^2) \\
&\leq \frac{8\kappa_{\max}}{N}\left(\sum_{j=1}^{N} f(x_{j-1}) - f(x_j)\right) \\
&\leq \frac{8\kappa_{\max}}{N}\left(f(x_0) - f^*\right).
\end{aligned}$$

Suppose the function f is convex. Using in the stopping criteria (3.17) in replacement of (3.11), we deduce a similar expression as (B.3):

$$\begin{aligned}
f(x_k) \leq \alpha_k f(x^*) + (1-\alpha_k)f(x_{k-1}) &+ \frac{\alpha_k^2\kappa_{\text{cvx}}}{2}\left(\|x^* - v_{k-1}\|^2 - \|x^* - v_k\|^2\right) \\
&- \frac{\kappa_{\text{cvx}}}{2}\|\tilde{x}_k - y_k\|^2 + \frac{\alpha_k\kappa_{\text{cvx}}}{k+1}\|\tilde{x}_k - y_k\|\,\|x^* - v_k\|.
\end{aligned}$$

Denote $\theta_k = \frac{1}{k+1}$. Completing the square, we obtain

$$\frac{-\kappa_{\text{cvx}}}{2}\|\tilde{x}_k - y_k\|^2 + \alpha_k\theta_k\kappa_{\text{cvx}}\|\tilde{x}_k - y_k\|\,\|x^* - v_k\| \leq \frac{\kappa_{\text{cvx}}}{2}(\alpha_k\theta_k)^2\|x^* - v_k\|^2.$$

Hence, we deduce

$$\begin{aligned}
f(x_k) - f^* &\leq (1-\alpha_k)(f(x_{k-1}) - f^*) + \frac{\alpha_k^2\kappa_{\text{cvx}}}{2}\left(\|x^* - v_{k-1}\|^2 - \|x^* - v_k\|^2\right) \\
&\qquad + \frac{\kappa_{\text{cvx}}}{2}(\alpha_k\theta_k)^2\|x^* - v_k\|^2. \\
&= (1-\alpha_k)(f(x_{k-1}) - f^*) + \frac{\alpha_k^2\kappa_{\text{cvx}}}{2}\left(\|x^* - v_{k-1}\|^2 - \left(1-\theta_k^2\right)\|x^* - v_k\|^2\right)
\end{aligned}$$

Denote $A_k := 1 - \theta_k^2$. Following the standard recursion argument as in the proofs of Theorem 3.3.2 and Theorem 3.3.2, we conclude

$$\begin{aligned}
\frac{f(x_k) - f^*}{\alpha_k^2} + \frac{A_k\kappa_{\text{cvx}}}{2}\|x^* - v_k\|^2 &\leq \frac{1-\alpha_k}{\alpha_k^2}\left(f(x_{k-1}) - f^*\right) + \frac{\kappa_{\text{cvx}}}{2}\|x^* - v_{k-1}\|^2 \\
&\leq \frac{1}{A_{k-1}}\left(\frac{f(x_{k-1}) - f^*}{\alpha_{k-1}^2} + \frac{A_{k-1}\kappa_{\text{cvx}}}{2}\|x^* - v_{k-1}\|^2\right).
\end{aligned}$$

The last inequality follows because $0 < A_{k-1} \leq 1$. Iterating N times, we deduce

$$\frac{f(x_N) - f^*}{\alpha_N^2} \leq \prod_{j=2}^{N}\frac{1}{A_{j-1}}\left(\frac{\kappa_{\text{cvx}}}{2}\|x^* - v_0\|^2\right). \tag{B.10}$$

We note

$$\prod_{j=2}^{N} \frac{1}{A_{j-1}} = \frac{1}{\prod_{j=2}^{N}\left(1 - \frac{1}{(j+1)^2}\right)} \leq 2;$$

thus the result is shown. Summing up (B.9) from $j = N+1$ to $2N$, we obtain

$$\begin{aligned} \min_{j=1,\ldots,2N} \left\{\operatorname{dist}^2(0, \partial f(\bar{x}_j))\right\} &\leq \frac{4}{N} \sum_{j=N+1}^{2N} \|\kappa_k(\bar{x}_k - x_{k-1})\|^2) \\ &\leq \frac{8\kappa_{\max}}{N} \left(\sum_{j=N+1}^{2N} f(x_{j-1}) - f(x_j) \right) \\ &\leq \frac{8\kappa_{\max}}{N} \left(f(x_N) - f^* \right) \end{aligned}$$

Combining this inequality with (B.10), the result is shown. □

B.4 Inner-loop complexity: proof of Theorem 3.4.4

Recall, the following notation

$$\begin{aligned} f_0(x; y) &= \frac{1}{n} \sum_{i=1}^{n} f_i(x) + \frac{\kappa}{2} \|x - y\|^2 \\ y^0 &= \operatorname{prox}_{1/(\kappa+L) f_0} \left(y - \frac{1}{\kappa + L} \nabla f_0(y; y) \right). \end{aligned} \tag{B.11}$$

Lemma B.4.1 (Relationship between function values and iterates of the prox)**.** *Assuming $\psi(x)$ is convex and the parameter $\kappa > \rho$, then*

$$f_\kappa(y^0; y) - f_\kappa^*(y) \leq \frac{\kappa + L}{2} \|y^* - y\|^2 \tag{B.12}$$

where y^* is a minima of $f_\kappa(\cdot; y)$ and $f_\kappa^*(y)$ is the optimal value.

Proof. As the κ is chosen sufficiently large, we know $f_0(\cdot; y)$ is convex and differentiable with $(\kappa + L)$-Lipschitz continuous gradient. Hence, we deduce for all x

$$f_0(y; y) + \nabla f_0(y; y)^T (x - y) \leq f_0(x; y). \tag{B.13}$$

Using the definition of y^0 and the $(\kappa + L)$-Lip. continuous gradient of $f_0(\cdot; y)$, we conclude for all x

$$\begin{aligned} f_\kappa(y^0; y) = f_0(y^0; y) + \psi(y^0) &\leq f_0(y; y) + \nabla f_0(y; y)^T (y_0 - y) + \frac{\kappa + L}{2} \|y_0 - y\|^2 + \psi(y_0) \\ &\leq f_0(y; y) + \nabla f_0(y; y)^T (x - y) + \frac{\kappa + L}{2} \|x - y\|^2 + \psi(x). \end{aligned} \tag{B.14}$$

By setting $x = y^*$ in both (B.13) and (B.14) and combining these results, we conclude

$$f_\kappa(y^0; y) \leq f_\kappa^*(y) + \frac{\kappa + L}{2} \|y^* - y\|^2 .$$

□

Note that if we are not in the composite setting and $\kappa > \rho$, then $f_\kappa(\cdot, y)$ is $(\kappa+L)$-strongly convex. Using standard bounds for strongly convex functions, Equation (B.12) follows (see [81]). We next show an important lemma for deducing the inner complexities.

Lemma B.4.2. *Assume $\kappa > \rho$. Given $\varepsilon \leq \frac{\kappa - \rho}{2}$, if an iterate z satisfies dist$(0, \partial f_\kappa(z; y)) \leq \varepsilon \|y^* - y\|$, then*

$$\textit{dist}(0, \partial f_\kappa(z; y)) \leq 2\varepsilon \|z - y\| . \tag{B.15}$$

Proof. Since $\kappa > \rho$, we know $f_\kappa(\cdot; y)$ is $(\kappa - \rho)$-strongly convex. Therefore, by [81], we know

$$\|z - y^*\| \leq \frac{1}{\kappa - \rho} \text{dist}(0, \partial f_\kappa(z; y)). \tag{B.16}$$

By the triangle inequality and Equation (B.16), we deduce

$$\begin{aligned} \text{dist}(0, \partial f_\kappa(z; y)) \leq \varepsilon \|y^* - y\| &\leq \varepsilon \big(\|y^* - z\| + \|z - y\| \big) \\ &\leq \frac{\varepsilon}{\kappa - \rho} \cdot \text{dist}(0, \partial f_\kappa(z; y)) + \varepsilon \|z - y\| \\ &\leq \frac{1}{2} \cdot \text{dist}(0, \partial f_\kappa(z; y)) + \varepsilon \|z - y\| . \end{aligned}$$

The last inequality follows because of the assumption $\varepsilon \leq \frac{\kappa - \rho}{2}$. Rearranging the terms above, we get the desired result. □

These two lemmas together give us Theorem 3.4.2.

Proof of Theorem 3.4.2. First, we prove that z_T satisfies both adaptive stationary condition and the descent condition. Recall, the point y^0 is defined to be the prox or y depending on if $f_\kappa(\cdot; y)$ is a composite form or smooth, respectively (see statement of Theorem 3.4.2). By Lemma B.4.1 (or the remark following it), the starting y^0 satisfies

$$f_\kappa(y^0; y) - f_\kappa^*(y) \leq \frac{\kappa + L}{2} \left\| y^* - y \right\|^2.$$

By the linear convergence assumption of $\mathcal{M}$ (see (3.14)) and the above equation, after $T := T_\kappa$ iterations initializing from y^0, we have

$$\begin{aligned} \operatorname{dist}^2(0, \partial f_\kappa(z_T; y)) &\leq A_\kappa (1 - \tau_\kappa)^T \left(f_\kappa(y^0; y) - f_\kappa^*(y) \right) \\ &\leq A_\kappa e^{-T \cdot \tau_\kappa} \left(f_\kappa(y^0; y) - f_\kappa^*(y) \right) \\ &\leq \frac{(\kappa - \rho)^2}{8(L + \kappa)} \cdot \frac{L + \kappa}{2} \left\| y^* - y \right\|^2 \\ &\leq \frac{(\kappa - \rho)^2}{16} \left\| y^* - y \right\|^2. \end{aligned} \tag{B.17}$$

Take the square root and apply Lemma B.4.2 yields

$$\operatorname{dist}(0, \partial f_\kappa(z_T; y)) \leq \frac{\kappa - \rho}{2} \left\| z_T - y \right\| \leq \kappa \left\| z_T - y \right\|,$$

which gives the adaptive stationary condition. Next, we show the descent condition. Let $v \in \partial f_\kappa(z_T; y)$ such that $\|v\| \leq (\kappa - \rho) \|z_T - y\| / 2$, by the $(\kappa - \rho)$-strong convexity of $f_\kappa(\cdot; y)$, we deduce

$$\begin{aligned} f_\kappa(y; y) &\geq f_\kappa(z_T; y) + \langle v, y - z_T \rangle + \frac{\kappa - \rho}{2} \left\| z_T - y \right\|^2 \\ &\geq f_\kappa(z_T; y) - \|v\| \left\| y - z_T \right\| + \frac{\kappa - \rho}{2} \left\| z_T - y \right\|^2 \\ &\geq f_\kappa(z_T; y). \end{aligned}$$

This yields the descent condition which completes the proof for T. The proof for S_κ is similar to T_κ, so we omit many of the details. In this case, we only need to show the adaptive stationary condition. For convenience, we denote $S = S_\kappa$. Following the same argument as in Equation (B.17) but with $S\log(k+1)$ number of iterations, we deduce

$$\text{dist}^2(0, \partial f_\kappa(z_S; y)) \leq \frac{(\kappa-\rho)^2}{16(k+1)^2} \left\| y^* - y \right\|^2.$$

By applying Lemma B.4.2, we obtain

$$\text{dist}(0, \partial f_\kappa(z_S; y)) \leq \frac{(\kappa-\rho)}{2(k+1)} \left\| z_T - y \right\| \leq \frac{\kappa}{k+1} \left\| z_S - y \right\|,$$

which proves the desired result for z_S. □

Assuming Proposition 3.4.5 and Proposition 3.4.6 hold as well as Lemma B.4.3, we begin by providing the proof of Theorem 3.4.4.

Proof of Theorem 3.4.4. We consider two cases: (i) the function f is non-convex and (ii) the function f is convex. First, we consider the non-convex setting. To produce $\bar{x}_k$, the method $\mathcal{M}$ is called

$$T \log\left(\tfrac{4L}{\kappa_0}\right) / \log(2) \tag{B.18}$$

number of times. This follows from Proposition 3.4.5 and Lemma B.4.3. The reasoning is that once $\kappa > \rho + L$, which only takes at most $\log(4L/\kappa_0)$ number of increases of κ to reach, then the iterate $\bar{x}_k$ satisfies the stopping criteria (3.18). Each time we increase κ we run $\mathcal{M}$ for T iterations. Therefore, the total number of iterations of $\mathcal{M}$ is given by multiplying T with $\log(4L/\kappa_0)$. To produce $\tilde{x}_k$, the method $\mathcal{M}$ is called $S\log(k+1)$ number of times. (Note: the proof of Theorem 3.4.3 does not need $\tilde{x}_k$ to satisfy (3.17) in the non-convex case).

Next, suppose the function f is convex. As before, to produce $\bar{x}_k$ the method $\mathcal{M}$ is called (B.18) times. To produce $\tilde{x}_k$, the method $\mathcal{M}$ is called $S\log(k+1)$ number of times. By Proposition 3.4.6, the iterate $\tilde{x}_k$ satisfies (3.17); a key ingredient in the proof of Theorem 3.4.3. □

B.4.1 Inner complexity for $\bar{x}_k$: proof of Proposition 3.4.5

Next, we supply the proof of Proposition 3.4.5 which shows that by choosing κ large enough, Algorithm 14 terminates.

Proof of Proposition 3.4.5. The idea is to apply Theorem 3.4.2. Since the parameter A_κ increases with κ, then we upper bound it by $A_{\kappa_k} \leq A_{4L}$. Moreover, we have $\kappa - \rho \geq \rho + L - \rho = L$. Lastly, since τ_κ is increasing in κ, we know $\frac{1}{\tau_\kappa} \leq \frac{1}{\tau_L}$. Plugging these bound into Theorem 3.4.2, we see that for any smoothing parameter κ satisfying $\rho + L < \kappa < 4L$, we get the desired result. □

Next, we compute the maximum number of times we must double κ until $\kappa > \rho + L$.

Lemma B.4.3 (Doubling κ)**.** *If we set T and S according to Theorem 3.4.4, then the doubling of κ_0 will terminate as soon as $\kappa > \rho + L$. Thus the number of times κ_0 must be doubled in Algorithm 14 is at most*

$$\frac{\log\left(\frac{2(\rho+L)}{\kappa_0}\right)}{\log(2)} \leq \left\lceil \frac{\log\left(\frac{4L}{\kappa_0}\right)}{\log(2)} \right\rceil .$$

Since κ is doubled (Algorithm 14) and T is chosen as in Proposition 3.4.5 , the maximum the value κ, $\kappa_{\max}$, takes is $2(\rho + L) \leq 4L$.

B.4.2 Inner complexity for $\tilde{x}_k$: proof of Proposition 3.4.6

In this section, we prove Proposition 3.4.6, an inner complexity result for the iterates $\tilde{x}_k$. Recall that the inner-complexity analysis for $\tilde{x}_k$ is important only when f is convex (see Section 3.4). Therefore, we assume throughout this section that the function f is convex. We are now ready to prove Proposition 3.4.6.

Proof of Proposition 3.4.6. The proof immediately follows from Theorem 3.4.2 by setting $\kappa = \kappa_{\text{cvx}}$ and $\rho = 0$ as the function f is convex. □

BIBLIOGRAPHY

[1] Z. Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *Preprint arXiv:1603.05953*, 2016.

[2] Z. Allen-Zhu. Natasha: Faster stochastic non-convex optimization via strongly non-convex parameter. *Preprint arXiv:1702.00763*, 2016.

[3] A. Aravkin, J.V. Burke, L. Ljung, A. Lozano, and G. Pilonetto. Generalized Kalman smoothing:modeling and algorithm. *Preprint arXiv:1609.06369*, 2016.

[4] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

[5] D. P. Bertsekas. *Nonlinear programming.* Athena scientific Belmont, 1999.

[6] D. P. Bertsekas. *Convex Optimization Algorithms.* Athena Scientific, 2015.

[7] J. Bolte, T.P. Nguyen, J. Peypouquet, and B. Suter. From error bounds to the complexity of first-order descent methods for convex functions. *Preprint arXiv:1510.08234*, 2015.

[8] J. Bolte and E. Pauwels. Majorization-minimization procedures and convergence of SQP methods for semi-algebraic and tame programs. *Math. Oper. Res.*, 41(2):442–465, 2016.

[9] J.M. Borwein and A.S. Lewis. *Convex Analysis and Nonlinear Optimization: Theory and Examples.* Springer, 2006.

[10] J.M. Borwein and Q.J. Zhu. *Techniques of Variational Analysis.* Springer Verlag, New York, 2005.

[11] G. Brassard and P. Bratley. *Fundamentals of algorithmics.* Prentice Hall, Inc., Englewood Cliffs, NJ, 1996.

[12] S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015.

[13] J.V. Burke. Descent methods for composite nondifferentiable optimization problems. *Math. Programming*, 33(3):260–279, 1985.

[14] J.V. Burke. An exact penalization viewpoint of constrained optimization. *SIAM J. Control Optim.*, 29(4):968–998, 1991.

[15] J.V. Burke, F.E. Curtis, H. Wang, and J. Wang. Iterative reweighted linear least squares for exact penalty subproblems on product sets. *SIAM J. Optim.*, 25(1):261–294, 2015.

[16] J.V. Burke and M.C. Ferris. A Gauss-Newton method for convex composite optimization. *Math. Programming*, 71(2, Ser. A):179–194, 1995.

[17] R.H. Byrd, J Nocedal, and R.A. Waltz. KNITRO: An integrated package for nonlinear optimization. In *Large-scale nonlinear optimization*, volume 83 of *Nonconvex Optim. Appl.*, pages 35–59. Springer, New York, 2006.

[18] Y. Carmon, J. C. Duchi, O. Hinder, and Aaron Sidford. Accelerated methods for non-convex optimization. *Preprint arXiv:1611.00756*, 2016.

[19] Y. Carmon, O. Hinder, J. C. Duchi, and A. Sidford. Convex until proven guilty: Dimension-free acceleration of gradient descent on non-convex functions. *Preprint arXiv:1705.02766*, 2017.

[20] C. Cartis, N.I.M. Gould, and P.L. Toint. On the evaluation complexity of composite function minimization with applications to nonconvex nonlinear programming. *SIAM J. Optim.*, 21(4):1721–1739, 2011.

[21] C. Cartis, N.I.M. Gould, and P.L. Toint. On the complexity of finding first-order critical points in constrained nonlinear optimization. *Mathematical Programming*, 2014.

[22] A. Cauchy. Méthode générale pour la résolution des systèm d'équations simultanées. *Compte Rendu des S'eances de L'Acad'emie des Sciences*, A(25):536–538, October 1847.

[23] D.I. Clark. The mathematical structure of Huber's M-estimator. *SIAM journal on scientific and statistical computing*, 6(1):209–219, 1985.

[24] F.H. Clarke, Y. Ledyaev, R.I. Stern, and P.R. Wolenski. *Nonsmooth Analysis and Control Theory.* Texts in Math. 178, Springer, New York, 1998.

[25] F.H. Clarke, R.J. Stern, and P.R. Wolenski. Proximal smoothness and the lower-C^2 property. *Journal of Convex Analysis*, 2(1-2):117–144, 1995.

[26] T.F. Coleman and A.R. Conn. Nonlinear programming via an exact penalty function: global analysis. *Math. Programming*, 24(2):137–161, 1982.

[27] A.R. Conn, K. Scheinberg, and L.N. Vicente. *Introduction to derivative-free optimization*, volume 8 of *MPS/SIAM Series on Optimization.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Programming Society (MPS), Philadelphia, PA, 2009.

[28] A. Daniilidis, D. Drusvyatskiy, and A.S. Lewis. Orthogonal invariance and identifiability. *SIAM J. Matrix Anal. Appl.*, 35(2):580–598, 2014.

[29] A. Daniilidis, A.S. Lewis, J. Malick, and H. Sendov. Prox-regularity of spectral functions and spectral sets. *J. Convex Anal.*, 15(3):547–560, 2008.

[30] A. Daniilidis and J. Malick. Filling the gap between lower-C^1 and lower-C^2 functions. *J. Convex Anal.*, 12(2):315–329, 2005.

[31] A. Daniilidis, J. Malick, and H.S. Sendov. Locally symmetric submanifolds lift to spectral manifolds. *Preprint U.A.B.* **23***/2009, 43 p., arXiv:1212.3936 [math.OC]*, 2012.

[32] C. Davis. All convex invariant functions of hermitian matrices. *Arch. Math.*, 8:276–278, 1957.

[33] A. J. Defazio, T. S. Caetano, and J. Domke. Finito: A faster, permutable incremental gradient method for big data problems. In *Proc. International Conference on Machine Learning (ICML)*, 2014.

[34] A.J. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[35] G. Di Pillo and L. Grippo. Exact penalty functions in constrained optimization. *SIAM J. Control Optim.*, 27(6):1333–1360, 1989.

[36] D. Drusvyatskiy, A.D. Ioffe, and A.S. Lewis. Nonsmooth optimization using taylor-like models: error bounds, convergence, and termination criteria. *Preprint arXiv:1610.03446*, 2016.

[37] D. Drusvyatskiy and M. Larsson. Approximating functions on stratified sets. *Trans. Amer. Math. Soc.*, 367(1):725–749, 2015.

[38] D. Drusvyatskiy and A.S. Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *Preprint arXiv:1602.06661*, 2016.

[39] D. Drusvyatskiy and C. Paquette. Efficiency of minimizing compositions of convex functions and smooth maps. *Preprint arXiv:1605.00125*, 2016.

[40] D. Drusvyatskiy and C. Paquette. Variational analysis of spectral functions simplified. *Journal of Convex Analysis (to appear)*, 2016.

[41] J.C. Duchi and F. Ruan. Solving (most) of a set of quadratic equalities: Composite optimization for robust phase retrieval. *Preprint arXiv:1705.02356*, 2017.

[42] J.C. Duchi and F. Ruan. Stochastic methods for composite optimization problems. *Preprint arXiv:1703.08570*, 2017.

[43] R. Dutter and P.J. Huber. Numerical methods for the nonlinear robust regression problem. *J. Statist. Comput. Simulation*, 13(2):79–113, 1981.

[44] I.I. Eremin. The penalty method in convex programming. *Cybernetics*, 3(4):53–56 (1971), 1967.

[45] H. Federer. Curvature measures. *Transactions of the American Mathematical Society*, 93:418–491, 1959.

[46] R. Fletcher. A model algorithm for composite nondifferentiable optimization problems. *Math. Programming Stud.*, (17):67–76, 1982. Nondifferential and variational techniques in optimization (Lexington, Ky., 1980).

[47] R. Frostig, R. Ge, S. M. Kakade, and A. Sidford. Un-regularizing: approximate proximal point algorithms for empirical risk minimization. In *Proc. International Conference on Machine Learning (ICML)*, 2015.

[48] S. Ghadimi and G. Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Math. Program.*, 156(1-2, Ser. A):59–99, 2016.

[49] S. Ghadimi, G. Lan, and H. Zhang. Generalized uniformly optimal methods for nonlinear programming. Technical report, Department of Industrial and Systems Engineering, University of Florida, 2015.

[50] N. Gillis. The why and how of nonnegative matrix factorization. In *Regularization, optimization, kernels, and support vector machines*, Chapman & Hall/CRC Mach. Learn. Pattern Recogn. Ser., pages 257–291. CRC Press, Boca Raton, FL, 2015.

[51] O. Güler. On the convergence of the proximal point algorithm for convex minimization. *SIAM Journal on Control and Optimization*, 29(2):403–419, 1991.

[52] T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical Learning With Sparsity: The Lasso And Generalizations*. CRC Press, 2015.

[53] J.-B. Hiriart-Urruty. ε-subdifferential calculus. In *Convex analysis and optimization (London, 1980)*, volume 57 of *Res. Notes in Math.*, pages 43–92. Pitman, Boston, Mass.-London, 1982.

[54] P. J. Huber. *Robust Statistics*. John Wiley and Sons, 2 edition, 2004.

[55] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[56] G. Lan. An optimal randomized incremental gradient method. *arXiv:1507.02000*, 2015.

[57] J.M. Lee. *Introduction to smooth manifolds*, volume 218 of *Graduate Texts in Mathematics*. Springer, New York, second edition, 2013.

[58] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quart. Appl. Math.*, 2:164–168, 1944.

[59] A.S. Lewis. Convex analysis on the Hermitian matrices. *SIAM J. Optim.*, 6(1):164–177, 1996.

[60] A.S. Lewis. Derivatives of spectral functions. *Math. Oper. Res.*, 21(3):576–588, 1996.

[61] A.S. Lewis. Nonsmooth analysis of eigenvalues. *Math. Program.*, 84(1, Ser. A):1–24, 1999.

[62] A.S. Lewis. Convex analysis on Cartan subspaces. *Nonlinear Anal.*, 42(5, Ser. A: Theory Methods):813–820, 2000.

[63] A.S. Lewis and H.S. Sendov. Twice differentiable spectral functions. *SIAM J. Matrix Anal. Appl.*, 23(2):368–386 (electronic), 2001.

[64] A.S. Lewis and H.S. Sendov. Nonsmooth analysis of singular values. I. Theory. *Set-Valued Anal.*, 13(3):213–241, 2005.

[65] A.S. Lewis and H.S. Sendov. Nonsmooth analysis of singular values. II. Applications. *Set-Valued Anal.*, 13(3):243–264, 2005.

[66] A.S. Lewis and S.J. Wright. A proximal method for composite minimization. *Math. Program.*, pages 1–46, 2015.

[67] H. Li and Z. Lin. Accelerated proximal gradient methods for nonconvex programming. In *Advances in Neural Information Processing Systems (NIPS)*. 2015.

[68] W. Li and J. Swetits. The linear l1 estimator and the huber m-estimator. *SIAM Journal on Optimization*, 8(2):457–475, 1998.

[69] H. Lin, J. Mairal, and Z. Harchaoui. A universal catalyst for first-order optimization. *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[70] H. Lin, J. Mairal, and Z. Harchaoui. QuickeNing: A generic Quasi-Newton algorithm for faster gradient-based optimization. *Preprint arXiv:1610.00960*, 2016.

[71] J. Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

[72] J. Mairal, F. Bach, and J. Ponce. Sparse modeling for image and vision processing. *Foundations and Trends in Computer Graphics and Vision*, 8(2-3):85–283, 2014.

[73] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *JMLR*, 11:19–60, 2010.

[74] D.W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Indust. Appl. Math.*, 11:431–441, 1963.

[75] B.S. Mordukhovich. *Variational Analysis and Generalized Differentiation I: Basic Theory.* Grundlehren der mathematischen Wissenschaften, Vol 330, Springer, Berlin, 2006.

[76] J.J. Moré. The Levenberg-Marquardt algorithm: implementation and theory. In *Numerical analysis (Proc. 7th Biennial Conf., Univ. Dundee, Dundee, 1977)*, pages 105–116. Lecture Notes in Math., Vol. 630. Springer, Berlin, 1978.

[77] S.C. Narula and J.F. Wellington. The minimum sum of absolute errors regression: a state of the art survey. *Internat. Statist. Rev.*, 50(3):317–326, 1982.

[78] A.S. Nemirovsky and D.B. Yudin. *Problem complexity and method efficiency in optimization.* A Wiley-Interscience Publication. John Wiley & Sons, Inc., New York, 1983. Translated from the Russian and with a preface by E. R. Dawson, Wiley-Interscience Series in Discrete Mathematics.

[79] Y. Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269(3):543–547, 1983.

[80] Y. Nesterov. On an approach to the construction of optimal methods of minimization of smooth convex functions. *Ekonom. i. Mat. Metody*, 24:125–161, 1988.

[81] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course.* Springer, 2004.

[82] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Prog.*, 2005.

[83] Y. Nesterov. Modified Gauss-Newton scheme with worst case guarantees for global performance. *Optim. Methods Softw.*, 22(3):469–483, 2007.

[84] Y. Nesterov. Primal-dual subgradient methods for convex problems. *Math. Program.*, 120(1):221–259, April 2009.

[85] Y. Nesterov. How to make the gradients small. *OPTIMA, MPS Newsletter*, (88):10–11, 2012.

[86] Y. Nesterov. Gradient methods for minimizing composite functions. *Math. Program.*, 140(1, Ser. B):125–161, 2013.

[87] J. Nocedal and S.J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.

[88] C. Paquette, H. Lin, D. Drusvyatskiy, J. Mairal, and Z. Harchaoui. Catalyst acceleration for gradient-based non-convex optimization. *Preprint arXiv:1703.10993*, 2017.

[89] N. Parikh and S.P. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2014.

[90] E. Pauwels. The value function approach to convergence analysis in composite optimization. *Oper. Res. Lett.*, 44(6):790–795, 2016.

[91] R.A. Poliquin and R.T. Rockafellar. Prox-regular functions in variational analysis. *Trans. Amer. Math. Soc.*, 348:1805–1838, 1996.

[92] M.J.D. Powell. General algorithms for discrete nonlinear approximation calculations. In *Approximation theory, IV (College Station, Tex., 1983)*, pages 187–218. Academic Press, New York, 1983.

[93] M.J.D. Powell. On the global convergence of trust region algorithms for unconstrained minimization. *Math. Programming*, 29(3):297–303, 1984.

[94] S.J. Reddi, S. Sra, B. Poczos, and A.J. Smola. Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[95] R.T. Rockafellar. *Convex Analysis.* Princeton University Press, 1970.

[96] R.T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

[97] R.T. Rockafellar. Favorable classes of Lipschitz-continuous functions in subgradient optimization. In *Progress in nondifferentiable optimization*, volume 8 of *IIASA Collaborative Proc. Ser. CP-82*, pages 125–143. Internat. Inst. Appl. Systems Anal., Laxenburg, 1982.

[98] R.T. Rockafellar and R. J.-B. Wets. *Variational analysis*, volume 317 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences].* Springer-Verlag, Berlin, 1998.

[99] M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 2016.

[100] M. Schmidt, Nicolas L.R., and Francis R.B. Convergence rates of inexact proximal-gradient methods for convex optimization. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1458–1466. Curran Associates, Inc., 2011.

[101] H.S. Sendov. *Variational spectral analysis.* ProQuest LLC, Ann Arbor, MI, 2001. Thesis (Ph.D.)–University of Waterloo (Canada).

[102] H.S. Sendov. The higher-order derivatives of spectral functions. *Linear Algebra Appl.*, 424(1):240–281, 2007.

[103] S. Shalev-Shwartz and T. Zhang. Proximal stochastic dual coordinate ascent. *arXiv:1211.2717*, 2012.

[104] S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, 2015.

[105] E. Siemsen and K.A. Bollen. Least absolute deviation estimation in structural equation modeling. *Sociol. Methods Res.*, 36(2):227–265, 2007.

[106] M. Šilhavý. Differentiability properties of isotropic functions. *Duke Math. J.*, 104(3):367–373, 2000.

[107] J. Sylvester. On the differentiability of O(n) invariant functions of symmetric matrices. *Duke Math. J.*, 52(2):475–483, 1985.

[108] C.M. Theobald. An inequality for the trace of the product of two symmetric matrices. *Math. Proc. Cambridge Philos. Soc.*, 77:265–267, 1975.

[109] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. *Technical Report*, 2008.

[110] S. Villa, S. Salzo, L. Baldassarre, and A. Verri. Accelerated and inexact forward-backward algorithms. *SIAM J. Optim.*, 23(3):1607–1633, 2013.

[111] J. von Neumann. Some matrix inequalities and metrization of matrix-space. *Tomck. Univ. Rev.*, 1:286–300, 1937.

[112] G.A. Watson. Characterization of the subdifferential of some matrix norms. *Linear Algebra and Its Applications*, pages 33–45, 1992.

[113] S.M. Wild. *Solving Derivative-Free Nonlinear Least Squares Problems with POUNDERS*. 2014. Argonne National Lab.

[114] B.E. Woodworth and N. Srebro. Tight complexity bounds for optimizing composite objectives. In *Advances in Neural Information Processing Systems (NIPS)*. 2016.

[115] S.J. Wright. Convergence of an inexact algorithm for composite nonsmooth optimization. *IMA J. Numer. Anal.*, 10(3):299–321, 1990.

[116] L. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

[117] Y. Yuan. On the superlinear convergence of a trust region algorithm for nonsmooth optimization. *Math. Programming*, 31(3):269–285, 1985.

[118] Y. Zhang and L. Xiao. Stochastic primal-dual coordinate method for regularized empirical risk minimization. In *Proc. International Conference on Machine Learning (ICML)*, 2015.

[119] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.