Acceleration Methods

Alexandre d'Aspremont
CNRS & Ecole Normale Supérieure, Paris
aspremon@ens.fr

Damien Scieur
Samsung SAIT AI Lab & Mila, Montreal
damien.scieur@gmail.com

Adrien Taylor
INRIA & Ecole Normale Supérieure, Paris
adrien.taylor@inria.fr

Last compiled: September 19, 2024

Typos and Errata

We have built a website to display all identified typos and errata.
You can visit it at: https://accelerationmethods.github.io/AccelerationMethodsWebsite/.
We encourage the reader to communicate any additional typos or errors to us.

Contents

ABSTRACT

This monograph covers some recent advances in a range of acceleration techniques frequently used in convex optimization. We first use quadratic optimization problems to introduce two key families of methods, namely momentum and nested optimization schemes. They coincide in the quadratic case to form the *Chebyshev method*.

We discuss momentum methods in detail, starting with the seminal work of Nesterov (1983) and structure convergence proofs using a few master templates, such as that for *optimized gradient methods*, which provide the key benefit of showing how momentum methods optimize convergence guarantees. We further cover proximal acceleration, at the heart of the *Catalyst* and *Accelerated Hybrid Proximal Extragradient* frameworks, using similar algorithmic patterns.

Common acceleration techniques rely directly on the knowledge of some of the regularity parameters in the problem at hand. We conclude by discussing *restart* schemes, a set of simple techniques for reaching nearly optimal convergence rates while adapting to unobserved regularity parameters.

1

Introduction

Optimization methods are a core component of the modern numerical toolkit. In many cases, iterative algorithms for solving convex optimization problems have reached a level of efficiency and reliability comparable to that of advanced linear algebra routines. This is largely true for medium scale-problems where interior point methods reign supreme, but less so for large-scale problems where the complexity of first-order methods is not as well understood and efficiency remains a concern.

The situation has improved markedly in recent years, driven in particular by the emergence of a number of applications in statistics, machine learning, and signal processing. Building on Nesterov's path-breaking algorithm from the 80's, several accelerated methods and numerical schemes have been developed that both improve the efficiency of optimization algorithms and refine their complexity bounds. Our objective in this monograph is to cover these recent developments using a few master templates.

The methods described in this manuscript can be arranged in roughly two categories. The first, stemming from the work of Nesterov (1983), produces variants of the gradient method with accelerated worst-case convergence rates that are provably optimal under classical regularity assumptions. The second uses outer iteration (a.k.a. nested) schemes to speed up convergence. In this second setting, accelerated schemes run both an inner loop and an outer loop, with the inner iterations being solved by classical optimization methods, and the outer loop containing the acceleration mechanism.

Direct acceleration techniques. Ever since the original algorithm by Nesterov (1983), the acceleration phenomenon was regarded as somewhat of a mystery. While accelerated gradient methods can be seen as iteratively building a model for the function and using it to guide gradient computations, the argument is essentially algebraic and is simply an effective exploitation of regularity assumptions. This approach of collecting inequalities

induced by regularity assumptions and cleverly chaining them to prove convergence was also used in e.g., (Beck and Teboulle, 2009a), to produce an optimal proximal gradient method. There too, however, the proof yielded little evidence as to why the method is actually faster.

Fortunately, we are now better equipped to push the proof mechanisms much further. Recent advances in the programmatic design of optimization algorithms allow us to design and analyze algorithms by following a more principled approach. In particular, the *performance estimation approach*, pioneered by Drori and Teboulle (2014), can be used to design optimal methods from scratch, selecting algorithmic parameters to optimize worst-case performance guarantees (Drori and Teboulle, 2014; Kim and Fessler, 2016). Primal dual optimality conditions on the design problem then provide a blueprint for the accelerated algorithm structure and for its convergence proof.

Using this framework, acceleration is no longer a mystery: it is the main objective in the design of the algorithm. We recover the usual "soup of regularity inequalities" that forms the template of classical convergence proofs, but the optimality conditions of the design problem explicitly produce a method that optimizes the convergence guarantee. In this monograph, we cover accelerated first-order methods using this systematic template and describe a number of convergence proofs for classical variants of the accelerated gradient method, such as those of Nesterov (1983; 2003), Beck and Teboulle (2009a) and Tseng (2008) as well as more recent ones (Kim and Fessler, 2016).

Nested acceleration schemes. The second category of acceleration techniques that we cover in this monograph is composed of outer iteration schemes, in which classical optimization algorithms are used as a black-box in the inner loop and acceleration is produced by an argument in the outer loop. We describe three acceleration results of this type.

The first scheme is based on nonlinear acceleration techniques. Based on arguments dating back to (Aitken, 1927; Wynn, 1956; Anderson and Nash, 1987), these techniques use a weighted average of iterates to extrapolate a better candidate solution than the last iterate. We begin by describing the Chebyshev method for solving quadratic problems, which interestingly qualifies both as a gradient method and as an outer iteration scheme. It takes its name from the use of Chebyshev polynomial coefficients to approximately minimize the gradient at the extrapolated solution. The argument can be extended to non-quadratic optimization problems provided the extrapolation procedure is regularized.

The second scheme, due to (Güler, 1992; Monteiro and Svaiter, 2013; Lin *et al.*, 2015) relies on a conceptual accelerated proximal point algorithm, and uses classical iterative methods to approximate the proximal point in an inner loop. In particular, this framework produces accelerated gradient methods (in the same sense as Nesterov's acceleration) when the approximate proximal points are computed using linearly converging gradient-based optimization methods, taking advantage of the fact that the inner problems are always strongly convex.

Finally, we describe restart schemes. These techniques exploit regularity properties

called Hölderian error bounds, which extend strong convexity properties near the optimum and hold almost generically, to improve the convergence rates of most first-order methods. The parameters of the Hölderian error bounds are usually unknown, but the restart schemes are robust: that is, they are adaptive to the Hölderian parameters and their empirical performance is excellent on problems with reasonable precision targets.

Content and organization. We present a few convergence acceleration techniques that are particularly relevant in the context of (first-order) convex optimization. Our summary includes our own points of view on the topic and is focused on techniques that have received substantial attention since the early 2000's, although some of the underlying ideas are much older. We do not pretend to be exhaustive, and we are aware that valuable references might not appear below.

The sections can be read nearly independently. However, we believe the insights of some sections can benefit the understanding of others. In particular, Chebyshev acceleration (Section 2) and nonlinear acceleration (Section 3) are clearly complementary readings. Similarly, Chebyshev acceleration (Section 2) and Nesterov acceleration (Section 4), Nesterov acceleration (Section 4) and proximal acceleration (Section 5), as well as Nesterov acceleration (Section 4) and restart schemes (Section 6) certainly belong together.

Prerequisites and complementary readings. This monograph is not meant to be a general-purpose manuscript on convex optimization, for which we refer the reader to the now classical references (Boyd and Vandenberghe, 2004; Bonnans *et al.*, 2006; Nocedal and Wright, 2006). Other directly related references are provided in the text.

We assume the reader to have a working knowledge of base linear algebra and convex analysis (such as of subdifferentials), as we do not detail the corresponding technical details while building on them. Classical references on the latter include (Rockafellar, 1970; Rockafellar and Wets, 2009; Hiriart-Urruty and Lemaréchal, 2013).

2

Chebyshev Acceleration

While "Chebyshev polynomials are everywhere dense in numerical analysis," we would like to argue here that Chebyshev polynomials also provide one of the most direct and intuitive explanations for acceleration arguments in first-order methods. That is, one can form linear combinations of past gradients for optimizing a worst-case guarantee on the distance to an optimal solution. In quadratic optimization, these linear combinations emerge from a Chebyshev minimization problem, whose solution can also be computed iteratively, thereby yielding an algorithm called the *Chebyshev method* (Nemirovsky and Polyak, 1984). The Chebyshev method traces its roots to at least Flanders and Shortley (1950), who credit Tuckey and Grosch. Its recurrence matches asymptotically the one of the heavy-ball method and is detailed below.

2.1 Introduction

In this section, we demonstrate basic acceleration results on quadratic minimization problems. In such problems, optimal points are the solutions of a linear system, and the basic gradient method can be seen as a simple iterative solver for this linear system. In this context, acceleration methods can be obtained using a classical argument involving Chebyshev polynomials.

Analyzing this simple scenario is useful in two ways. First, recursive formulations of the Chebyshev argument yield a basic algorithmic template for designing accelerated methods and provide first approach to their structures, such as the presence of a momentum term. Second, the arguments are robust to perturbations of the quadratic function f and hence apply in more generic contexts. This property enables acceleration in a wider range of applications, which we cover later in the Section 3 and Section 4.

For now, consider the following unconstrained quadratic minimization problem

$$\text{minimize} \quad \left\{ f(x) \triangleq \frac{1}{2}\langle x; \mathbf{H}x \rangle - \langle b; x \rangle \right\} \tag{2.1}$$

in the variable $x \in \mathbb{R}^d$, where $\mathbf{H} \in \mathbf{S}_d$ (the set of symmetric matrices of size $d \times d$) is the Hessian of f. We further assume that f is both smooth and strongly convex, i.e., that there exist some $L > \mu > 0$ such that $\mu\mathbf{I} \preceq \mathbf{H} \preceq L\mathbf{I}$. The reasoning of this section readily extends to the case where μ is the smallest *nonzero* eigenvalue of $\mathbf{H}$. We start by analyzing the convergence of the fixed step gradient method (Algorithm 1) for solving (2.1).

Algorithm 1 Gradient method

Input: A differentiable convex function f, initial point x_0, step size $\gamma > 0$, budget N.

1: **for** $k = 1, \ldots, N$ **do**
2: $\quad x_k = x_{k-1} - \gamma \nabla f(x_{k-1})$
3: **end for**

Output: Approximate solution x_N.

For problem (2.1), the iteration reads

$$x_{k+1} = (\mathbf{I} - \gamma\mathbf{H})x_k + \gamma b,$$

and calling $x_\star$ the optimum of problem (2.1) (satisfying $\mathbf{H}x_\star = b$) yields

$$x_{k+1} - x_\star = (\mathbf{I} - \gamma\mathbf{H})(x_k - x_\star). \tag{2.2}$$

This means the iterates of gradient descent $x_k - x_\star$ can be computed from $x_0 - x_\star$ via $x_k - x_\star = P_k^{\text{Grad}}(\mathbf{H})(x_0 - x_\star)$ using the matrix polynomial

$$P_k^{\text{Grad}}(\mathbf{H}) = (\mathbf{I} - \gamma\mathbf{H})^k. \tag{2.3}$$

Suppose we set the step size γ to ensure

$$\|\mathbf{I} - \gamma\mathbf{H}\|_2 < 1,$$

where $\|\cdot\|_2$ stands for the operator ℓ_2 norm. Then, (2.2) controls the convergence with

$$\|x_k - x_\star\|_2 \leq \|\mathbf{I} - \gamma\mathbf{H}\|_2^k \, \|x_0 - x_\star\|_2, \quad \text{for } k \geq 0. \tag{2.4}$$

Because the matrix $\mathbf{H}$ is symmetric and hence diagonalizable in an orthogonal basis, given $\gamma > 0$, we obtain

$$\begin{aligned}
\|\mathbf{I} - \gamma\mathbf{H}\|_2 &\leq \max_{\mu\mathbf{I} \preceq H \preceq L\mathbf{I}} \|\mathbf{I} - \gamma\mathbf{H}\|_2 \\
&\leq \max_{\mu \leq \lambda \leq L} |1 - \gamma\lambda| \\
&\leq \max_{\mu \leq \lambda \leq L} \max\{\gamma\lambda - 1 \,;\, 1 - \gamma\lambda\} \\
&\leq \max\{\gamma L - 1 \,;\, 1 - \gamma\mu\}.
\end{aligned}$$

To get the best possible worst-case convergence rate, we now minimize this quantity in γ by solving

$$\min_\gamma \ \max\{\gamma L - 1, 1 - \gamma\mu\} = \frac{L - \mu}{L + \mu}. \tag{2.5}$$

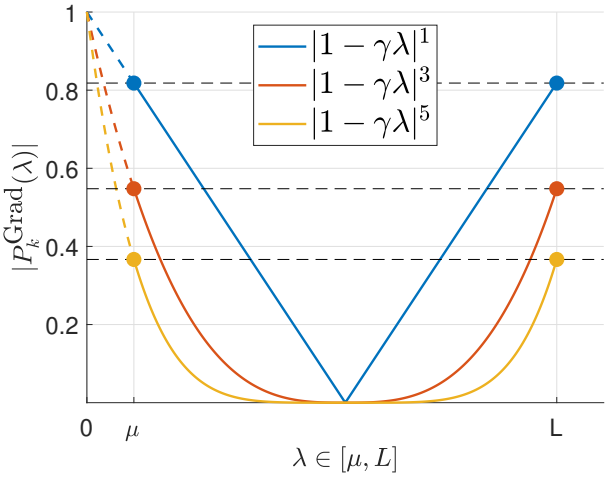

Figure 2.1: We plot $|P_k^{\text{Grad}}(\lambda)|$ (for the optimal γ in (2.6)) for $k \in \{1, 3, 5\}$, $\mu = 1$, $L = 10$. Note that the polynomials satisfy $P_k^{\text{Grad}}(0) = 1$. The rate is equal to the largest value of $|P_k^{\text{Grad}}(\lambda)|$ on the interval, which is achieved at the boundaries (where λ is either equal to μ or to L).

The optimal step size is obtained when both terms in the max are equal, reaching:

$$\gamma = \frac{2}{L+\mu}. \tag{2.6}$$

Denoting by $\kappa \triangleq \frac{L}{\mu} \geq 1$ the *condition number* of the function f, the bound in (2.4) finally becomes

$$\|x_k - x_\star\|_2 \leq \left(\frac{\kappa-1}{\kappa+1}\right)^k \|x_0 - x_\star\|_2, \quad \text{for } k \geq 0, \tag{2.7}$$

which is a worst-case guarantee for the gradient method when minimizing smooth strongly convex quadratic functions.

2.2 Optimal Methods and Minimax Polynomials

In Equation (2.4) above, we saw that the worst-case convergence rate of the gradient method on quadratic functions can be controlled by the spectral norm of a matrix polynomial. Figure 2.1 plots the polynomial P_k^{Grad} for several degrees k. We can extend this reasoning further to produce methods with accelerated worst-case convergence guarantees.

2.2.1 First-Order Methods and Matrix Polynomials

The bounds derived above for gradient descent can be extended to a broader class of first-order methods for quadratic optimization. We consider first-order algorithms in which each iterate belongs to the span of previous gradients, i.e.

$$x_{k+1} \in x_0 + \text{span}\left\{\nabla f(x_0), \nabla f(x_1), \ldots, \nabla f(x_k)\right\}, \tag{2.8}$$

and show that the iterates can be written using matrix polynomials as in (2.3) above.

Proposition 2.1. Let $x_0 \in \mathbb{R}^d$ and f be a quadratic function defined as in (2.1) with $\mu\mathbf{I} \preceq \mathbf{H} \preceq L\mathbf{I}$ for some $L > \mu > 0$. The sequence $\{x_k\}_{k=0,1,\dots}$ satisfies

$$x_{k+1} \in x_0 + \operatorname{span}\{\nabla f(x_0), \nabla f(x_1), \dots, \nabla f(x_k)\}, \tag{2.9}$$

for all $k = 0, 1, \dots$, if and only if the errors $\{x_k - x_\star\}_{k=0,1,\dots}$ can be written as

$$x_k - x_\star = P_k(\mathbf{H})(x_0 - x_\star), \tag{2.10}$$

for all $k = 0, 1 \dots$, for some sequence of polynomials $\{P_k\}_{k=0,1,\dots}$ with P_k of degree at most k and $P_k(0) = 1$.

Proof. Since $\nabla f(x)$ is the gradient of a quadratic function, it reads

$$\nabla f(x) = \mathbf{H}x - b = \mathbf{H}(x - x_\star)$$

for any $x_\star$ satisfying $\mathbf{H}x_\star = b$, where $\mathbf{H}$ is symmetric. We have

$$\begin{aligned} x_0 - x_\star &= 1 \cdot (x_0 - x_\star) \\ &= P_0(\mathbf{H})(x_0 - x_\star). \end{aligned}$$

We now show recursively that $x_k - x_\star = P_k(\mathbf{H})(x_0 - x_\star)$, where P_k is a residual polynomial of degree at most k. Our assumption about the iterates (2.9) implies that, for some sequence of coefficients $\{\alpha_i^{(k+1)}\}_{i=0,\dots,k}$,

$$x_{k+1} - x_\star = x_0 - x_\star + \sum_{i=0}^{k} \alpha_i^{(k+1)} \nabla f(x_i).$$

Assuming recursively that (2.10) holds for all indices $i \le k$,

$$\begin{aligned} x_{k+1} - x_\star &= x_0 - x_\star + \sum_{i=0}^{k} \alpha_i^{(k+1)} \mathbf{H} P_i(\mathbf{H})(x_0 - x_\star) \\ &= \left(\mathbf{I} + \mathbf{H} \sum_{i=0}^{k} \alpha_i^{(k+1)} P_i(\mathbf{H}) \right) (x_0 - x_\star). \end{aligned}$$

Then, by writing $P_{k+1}(x) = 1 + x \sum_{i=0}^{k} \alpha_i^{(k+1)} P_i(x)$, we have

$$x_{k+1} - x_\star = P_{k+1}(\mathbf{H})(x_0 - x_\star)$$

with $P_{k+1}(0) = 1$ and $\deg(P_{k+1}) \le k + 1$. Since the proof is a sequence of equalities, the equivalence readily follows. ∎

Given a class $\mathcal{M}$ of problem matrices $\mathbf{H}$, Proposition 2.1 provides a way to design algorithms. Indeed, we can extract a first-order method from a sequence of polynomials $\{P_k\}_{k=0,\dots,N}$. We can therefore use tools from approximation theory to find optimal polynomials and extract corresponding methods from them. Given a matrix class $\mathcal{M}$, this involves minimizing the worst-case convergence bound over $\mathbf{H} \in \mathcal{M}$ by solving

$$P_k^* = \operatorname*{argmin}_{\substack{P \in \mathcal{P}_k, \\ P(0)=1}} \max_{\mathbf{H} \in \mathcal{M}} \|P(\mathbf{H})\|_2 \tag{2.11}$$

where P_k is the set of polynomials of degree at most k. The polynomial P_k^* is an optimal polynomial for $\mathcal{M}$ and yields a (worst-case) optimal algorithm for the class $\mathcal{M}$. In terms of the notation used in the proof of Proposition 2.1, we are by construction looking for coefficients $\{\alpha_i^{(j)}\}_i$ depending only on the problem class $\mathcal{M}$, but *not* on a specific instance of $\mathbf{H}$.

2.3 The Chebyshev Method

In the case where $\mathcal{M}$ is the set of positive definite matrices with a bounded spectrum, namely

$$\mathcal{M} = \{\mathbf{H} \in \mathbf{S}_d : 0 \prec \mu\mathbf{I} \preceq \mathbf{H} \preceq L\mathbf{I}\},$$

the optimal polynomial can be found by solving

$$P_k^* = \underset{\substack{P \in \mathcal{P}_k, \\ P(0)=1}}{\operatorname{argmin}} \max_{\lambda \in [\mu, L]} |P(\lambda)| \tag{2.12}$$

Polynomials that solve (2.12) are derived from *Chebyshev polynomials of the first kind* in approximation theory and can be formed explicitly to produce an optimal algorithm called the *Chebyshev method*. This section describes this method and provides its corresponding worst-case convergence guarantees.

2.3.1 Shifted Chebyshev Polynomials

We now explicitly introduce the Chebyshev polynomials. A more complete treatment of these polynomials is available in, e.g., Mason and Handscomb (2002). Chebyshev polynomials of the first kind are defined recursively as follows

$$\begin{aligned} &\mathcal{T}_0(x) = 1, \\ &\mathcal{T}_1(x) = x, \\ &\mathcal{T}_k(x) = 2x\mathcal{T}_{k-1}(x) - \mathcal{T}_{k-2}(x), \quad \text{for } k \geq 2. \end{aligned} \tag{2.13}$$

There exists a compact explicit solution for Chebyshev polynomials that involves trigonometric functions:

$$\mathcal{T}_k(x) = \begin{cases} \cos(k \operatorname{acos}(x)) & x \in [-1, 1], \\ \cosh(k \operatorname{acosh}(x)) & x > 1, \\ (-1)^k \cosh(k \operatorname{acosh}(-x)) & x < 1. \end{cases} \tag{2.14}$$

It is possible to show that Chebyshev polynomials satisfy the minimax property

$$\frac{1}{2^{k-1}}\mathcal{T}_k = \underset{\substack{\deg(P) \leq k \\ P \text{ is monic}}}{\operatorname{argmin}} \max_{[-1,1]} |P(x)|,$$

where a monic polynomial is a polynomial whose coefficient associated with the highest power is equal to one. From this minimax definition, that defines the minimal polynomial

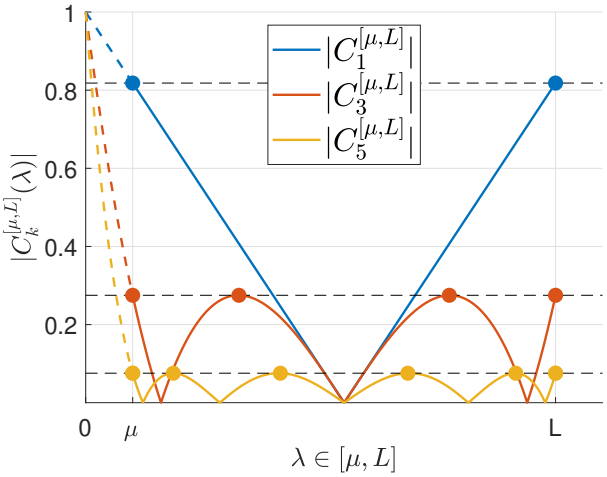

Figure 2.2: We plot the absolute value of $C_1^{[\mu,L]}(x)$, $C_3^{[\mu,L]}(x)$ and $C_5^{[\mu,L]}(x)$ for $\lambda \in [\mu, L]$, where $\mu = 1$ and $L = 10$. Note that the polynomials satisfy $C_k^{[\mu,L]}(0) = 1$. The maximum value of the image of $[\mu, L]$ by $C_k^{[\mu,L]}$ decreases rapidly as k grows, implying an accelerated rate of convergence.

over $[-1, 1]$, we can transform it by *shifting* it to the interval $[\mu, L]$, then *rescaling* it to obtain a polynomial such that $P(0) = 1$. More precisely, using a simple linear mapping from $[\mu,\ L]$ to $[-1,\ 1]$,

$$x \to t^{[\mu,L]}(x) = \frac{2x - (L + \mu)}{L - \mu},$$

we obtain *shifted Chebyshev polynomials*:

$$C_k^{[\mu,L]}(x) = \frac{\mathcal{T}_k(t^{[\mu,L]}(x))}{\mathcal{T}_k(t^{[\mu,L]}(0))}. \tag{2.15}$$

where we have enforced the normalization constraint $C_k^{[\mu,L]}(0) = 1$. Under these transformations, the shifted Chebyshev polynomials keep some minimax property, and can be shown to be solutions to (2.12).

More formally, Golub and Varga (1961a) characterize the ℓ_∞ optimality on the interval $[\mu, L]$ using an *equi-oscillation* argument (see, e.g., (Süli and Mayers, 2003)); i.e., they show that the solution of (2.12) is $P_k^* = C_k^{[\mu,L]}$. The equioscillation property of the shifted Chebyshev polynomial is clearly visible on Figure 2.2, where the polynomial hits its maximum value $k + 1$ times on the interval $[\mu, L]$.

2.3.2 Chebyshev Algorithm

The following recursion follows from (2.13) together with (2.15) and a few simplifications:

$$\begin{aligned} C_0^{[\mu,L]}(x) &= 1, \\ C_1^{[\mu,L]}(x) &= 1 - \frac{2}{L+\mu}x, \\ C_k^{[\mu,L]}(x) &= \frac{2\delta_k}{L-\mu}\left(L+\mu-2x\right)C_{k-1}^{[\mu,L]}(x) \\ &\quad + \left(1 - \frac{2\delta_k(L+\mu)}{L-\mu}\right)C_{k-2}^{[\mu,L]}(x), \quad \text{for } k \geq 2, \end{aligned} \tag{2.16}$$

where $\delta_1 = \frac{L-\mu}{L+\mu}$ and

$$\delta_k = -\frac{\mathcal{T}_{k-1}(t^{[\mu,L]}(0))}{\mathcal{T}_k(t^{[\mu,L]}(0))} = \frac{1}{2\frac{L+\mu}{L-\mu} - \delta_{k-1}}, \quad \text{for } k \geq 2.$$

The sequence δ_k ensures $C_k^{[\mu,L]}(0) = 1$. We present this recursion for simplicity, but one should note that it might present some numerical stability issues in practice. There exist numerically more stable first-order methods based on Chebyshev polynomials, see, e.g., (Gutknecht and Röllin, 2002, Algorithm 1). We plot $C_1^{[\mu,L]}(x)$, $C_3^{[\mu,L]}(x)$ and $C_5^{[\mu,L]}(x)$ in Figure 2.2 for illustration.

Computational details for the shifted Chebyshev. Let us quickly detail how to arrive to (2.16) using (2.13) and (2.15). We first expand $\mathcal{T}_k(t^{[\mu,L]}(x))$ using (2.13),

$$\begin{aligned} \mathcal{T}_k(t^{[\mu,L]}(x)) &= 2t^{[\mu,L]}(x)\mathcal{T}_{k-1}(t^{[\mu,L]}(x)) - \mathcal{T}_{k-2}(t^{[\mu,L]}(x)), \\ &= \frac{2(2x-L-\mu)}{L-\mu}\mathcal{T}_{k-1}(t^{[\mu,L]}(x)) - \mathcal{T}_{k-2}(t^{[\mu,L]}(x)). \end{aligned}$$

Since $C_k^{[\mu,L]}(x) = \frac{\mathcal{T}_k\left(t^{[\mu,L]}(x)\right)}{\mathcal{T}_k\left(t^{[\mu,L]}(0)\right)}$, we substitute $\mathcal{T}_k(t^{[\mu,L]}(x))$ by $\mathcal{T}_k(t^{[\mu,L]}(0)) \cdot C_k^{[\mu,L]}(x)$ in the equation above and obtain

$$\begin{aligned} \mathcal{T}_k(t^{[\mu,L]}(0)) \cdot C_k^{[\mu,L]}(x) =& \frac{2(2x-L-\mu)}{L-\mu}\mathcal{T}_{k-1}(t^{[\mu,L]}(0)) \cdot C_{k-1}^{[\mu,L]}(x) \\ & - \mathcal{T}_{k-2}(t^{[\mu,L]}(0)) \cdot C_{k-2}^{[\mu,L]}(x), \\ C_k^{[\mu,L]}(x) =& \frac{2(2x-L-\mu)}{L-\mu}\frac{\mathcal{T}_{k-1}(t^{[\mu,L]}(0))}{\mathcal{T}_k(t^{[\mu,L]}(0))} \cdot C_{k-1}^{[\mu,L]}(x) \\ & - \frac{\mathcal{T}_{k-2}(t^{[\mu,L]}(0))}{\mathcal{T}_k(t^{[\mu,L]}(0))} \cdot C_{k-2}^{[\mu,L]}(x), \\ C_k^{[\mu,L]}(x) =& -\frac{2(2x-L-\mu)}{L-\mu}\delta_k C_{k-1}^{[\mu,L]}(x) \\ & - \delta_{k-1}\delta_k C_{k-2}^{[\mu,L]}(x). \end{aligned}$$

For obtaining a simple recursion on δ_k, note that $C_k^{[\mu,L]}(0) = 1$ for all $k \geq 0$ by construction. It follows that

$$C_k^{[\mu,L]}(0) = \frac{2(L+\mu)}{L-\mu}\delta_k \underbrace{C_{k-1}^{[\mu,L]}(0)}_{=1} - \delta_{k-1}\delta_k \underbrace{C_{k-2}^{[\mu,L]}(0)}_{=1} = 1,$$

and thereby

$$\delta_{k-1}\delta_k = 1 - 2\delta_k \frac{L+\mu}{L-\mu} \quad \text{and} \quad \delta_k = \frac{1}{2\frac{L+\mu}{L-\mu} - \delta_{k-1}}.$$

2.3.3 Chebyshev and Polyak's Heavy-Ball Methods

We now present the resulting algorithm, called *Chebyshev semi-iterative method* (Golub and Varga, 1961b). We define iterates using $C_k^{[\mu,L]}(x)$ as follows,

$$x_k - x_\star = C_k^{[\mu,L]}(\mathbf{H})(x_0 - x_\star).$$

The recursion in (2.16) then yields

$$\begin{aligned} x_k - x_\star &= \frac{2\delta_k}{L-\mu}\left((L+\mu)\mathbf{I} - 2\mathbf{H}\right)(x_{k-1} - x_\star) \\ &\quad + \left(1 - 2\delta_k\frac{L+\mu}{L-\mu}\right)(x_{k-2} - x_\star). \end{aligned}$$

Since the gradient of the function reads

$$\nabla f(x_k) = \mathbf{H}(x_k - x_\star),$$

we can simplify away $x_\star$ to get the following recursion:

$$x_k = \frac{2\delta_k}{L-\mu}\left((L+\mu)x_{k-1} - 2\nabla f(x_{k-1})\right) + \left(1 - 2\delta_k\frac{L+\mu}{L-\mu}\right)x_{k-2},$$

which describes iterates of the Chebyshev method. We summarize it as Algorithm 2. By construction, the Chebyshev method is a worst-case optimal first-order method for minimizing quadratics whose spectrum lies in $[\mu, L]$. Surprisingly, its iteration structure is simple and somewhat intuitive: it involves a gradient step with variable step size $4\delta_k/(L-\mu)$, combined with a variable momentum term.

Perhaps more surprisingly, the Chebyshev method has a *stationary regime* that is even simpler. Indeed, when $k \to \infty$, the coefficients of the recursion from Algorithm 2 converge to the ones of *Polyak's heavy-ball method*,

$$x_k = x_{k-1} - \frac{4}{(\sqrt{L}+\sqrt{\mu})^2}\nabla f(x_{k-1}) + \frac{(\sqrt{L}-\sqrt{\mu})^2}{(\sqrt{L}+\sqrt{\mu})^2}(x_{k-1} - x_{k-2}).$$

To see this, it suffices to compute the limit of δ_k, written as δ_∞, by solving

$$\delta_\infty = \frac{1}{2\frac{L+\mu}{L-\mu} - \delta_\infty},$$

Algorithm 2 Chebyshev's method

Input: An L-smooth μ-strongly convex quadratic f, initial point x_0 and budget N.

1: Set $\delta_1 = \frac{L-\mu}{L+\mu}$, $x_1 = x_0 - \frac{2}{L+\mu}\nabla f(x_0)$.
2: **for** $k = 2, \dots, N$ **do**
3: Set $\delta_k = \frac{1}{2\frac{L+\mu}{L-\mu} - \delta_{k-1}}$,
4: $x_k = x_{k-1} - \frac{4\delta_k}{L-\mu}\nabla f(x_{k-1}) + \left(1 - 2\delta_k \frac{L+\mu}{L-\mu}\right)(x_{k-2} - x_{k-1})$.
5: **end for**

Output: Approximate solution x_N.

reaching

$$\delta_\infty = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}. \tag{2.17}$$

We obtain Polyak's heavy-ball method by replacing δ_k with δ_∞ in Algorithm 2.

2.3.4 Worst-case Convergence Bounds

The *shifted* Chebyshev polynomials are solutions to (2.12). Therefore, using the same trick as for gradient descent, we can obtain the following worst-case bound for the Chebyshev method

$$\|x_k - x_\star\|_2 \leq \|C_k^{[\mu,L]}(\mathbf{H})(x_0 - x_\star)\|_2 \leq \|x_0 - x_\star\|_2 \max_{x \in [\mu,L]} |C_k^{[\mu,L]}(x)|. \tag{2.18}$$

The maximum value is determined by evaluating the polynomial at one of the extremities of the interval (Mason and Handscomb, 2002, Chapter 2) (see also Figure 2.2), i.e,

$$\max_{x \in [\mu,L]} |C_k^{[\mu,L]}(x)| = C_k^{[\mu,L]}(L).$$

Using (2.15) and (2.14) successively,

$$|C_k^{[\mu,L]}(L)| = \frac{1}{|\mathcal{T}_k(t^{[\mu,L]}(0))|} = \frac{1}{\cosh\left(k \operatorname{acosh}\left(\frac{L+\mu}{L-\mu}\right)\right)}.$$

We obtain the worst-case convergence guarantee of the Chebyshev method, as stated in the following theorem.

Theorem 2.1. Let $x_0 \in \mathbb{R}^d$ and f be a quadratic function defined as in (2.1) with $\mu\mathbf{I} \preceq \mathbf{H} \preceq L\mathbf{I}$ for some $L > \mu > 0$. For any $N \in \mathbb{N}$, the iterates of the Chebyshev method (Algorithm 2) satisfy

$$\|x_N - x_\star\|_2 \leq \frac{2}{\xi^N + \xi^{-N}}\|x_0 - x_\star\|_2 \quad \text{where} \quad \xi = \frac{\sqrt{\frac{L}{\mu}} + 1}{\sqrt{\frac{L}{\mu}} - 1}. \tag{2.19}$$

Proof. We bound (2.18) as follows,

$$\begin{aligned}\|x_N - x_\star\|_2 &\leq \|x_0 - x_\star\|_2 \max_{x\in[\mu,L]} |C_N^{[\mu,L]}(x)| \\ &= \|x_0 - x_\star\|_2 \frac{1}{\cosh\left(N \operatorname{acosh}\left(\frac{L+\mu}{L-\mu}\right)\right)}.\end{aligned}$$

First, we evaluate the acosh term,

$$\begin{aligned}\operatorname{acosh}\left(\frac{L+\mu}{L-\mu}\right) &= \ln\left(\frac{L+\mu}{L-\mu} + \sqrt{\left(\frac{L+\mu}{L-\mu}\right)^2 - 1}\right), \\ &= \ln(\xi), \quad \text{where } \xi = \frac{\sqrt{\frac{L}{\mu}}+1}{\sqrt{\frac{L}{\mu}}-1}.\end{aligned}$$

After plugging this result into the cosh, we get

$$\frac{1}{\cosh(N\ln(\xi))} = \frac{2}{e^{N\ln(\xi)} + e^{-N\ln(\xi)}} = \frac{2}{\xi^N + \xi^{-N}},$$

thereby reaching the desired result. ∎

It may be difficult to compare the convergence rate of the Chebyshev method with that of gradient descent, due to its more complex expression. However, by neglecting the denominator term ξ^{-N}, we obtain the following upper bound:

$$\|x_N - x_\star\|_2 \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^N \|x_0 - x_\star\|_2.$$

Note that the convergence rate of Polyak's heavy-ball method matches (up to a multiplicative factor) that of Chebyshev's method asymptotically, which is better than that of gradient descent in (2.7), which reads

$$\|x_N - x_\star\|_2 \leq \left(\frac{\kappa-1}{\kappa+1}\right)^N \|x_0 - x_\star\|_2.$$

We summarize this result in the following corollary, which compares the number of iterations required to reach a target accuracy ϵ.

Corollary 2.2. Let $x_0 \in \mathbb{R}^d$ and f be a quadratic function defined as in (2.1) with $\mu\mathbf{I} \preceq \mathbf{H} \preceq L\mathbf{I}$ for some $L > \mu > 0$. After respectively

- $N \geq \frac{L}{2\mu} \log\left(\frac{\|x_0-x_\star\|_2}{\epsilon}\right)$ iterations of gradient descent (Algorithm 1 with (2.6)), or
- $N \geq \sqrt{\frac{L}{2\mu}} \log\left(\frac{\|x_0-x_\star\|_2}{\epsilon}\right)$ iterations of Chebyshev's method (Algorithm 2),

we have that

$$\|x_N - x_\star\|_2 \leq \epsilon.$$

Proof. For gradient descent, a sufficient condition on the number of iterations required to reach an accuracy of ϵ reads

$$\|x_N - x_\star\|_2 \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^N \|x_0 - x_\star\|_2 \leq \epsilon.$$

Taking the log on both sides, we get

$$N \geq \frac{\log\left(\frac{\|x_0 - x_\star\|_2}{\epsilon}\right)}{\log\left(\frac{\kappa+1}{\kappa-1}\right)}.$$

Using the bound $\log\left(\frac{\frac{1}{x}+1}{\frac{1}{x}-1}\right) > 2x$, the above condition can be simplified to the following stronger condition on N

$$N \geq \frac{\kappa}{2} \log\left(\frac{\|x_0 - x_\star\|_2}{\epsilon}\right).$$

This gives the desired result for gradient descent. With the same approach, we also get the result for the Chebyshev algorithm. ∎

This corollary shows that the Chebyshev method can be $\sqrt{\kappa}$ *faster* than gradient descent. This translate to a speedup factor of 100 in problems with a (reasonable) condition number of 10^4, which is very significant.

Worst-case optimality of Chebyshev's method. When the dimension d of the ambient space is sufficiently large, and without further assumptions on the spectrum of $\mathbf{H}$, the worst-case guarantee on Chebyshev's method is essentially unimprovable. Informally, given a budget N, a problem class $\mathcal{M}$ and some $R > 0$, the best worst-case guarantee on the distance to optimality $\|x_N - x_\star\|_2$ that can be achieved by a first-order method is given by

$$\max_{\substack{\mathbf{H}\in\mathcal{M},\, x_\star, x_0 \in \mathbb{R}^d \\ \|x_0 - x_\star\|_2 \leq R}} \min_{\substack{P\in\mathcal{P}_N, \\ P(0)=1}} \|P(\mathbf{H})(x_0 - x_\star)\|_2, \tag{2.20}$$

which corresponds to the worst-case performance of the best performing method on any problem of the class. More precisely, for any first-order method satisfying $x_k \in x_0 + \text{span}\{\nabla f(x_0), \nabla f(x_1), \ldots, \nabla f(x_{k-1})\}$ for all $k \geq 1$ (the "span assumption") applied on the quadratic problem (2.1), it holds that:

$$\begin{aligned} &\text{span}\{\nabla f(x_0), \ldots, \nabla f(x_{k-1})\} \\ &\qquad \subseteq \text{span}\left\{H(x_0 - x_\star), \ldots, H^k(x_0 - x_\star)\right\}, \end{aligned}$$

and therefore, the conjugate gradient-like method

$$\begin{aligned} x_k = \underset{x}{\text{argmin}}\ &\|x - x_\star\|_2 \\ &\text{s.t. } x \in x_0 + \text{span}\{H(x_0 - x_\star), \ldots, H^k(x_0 - x_\star)\}, \end{aligned} \tag{2.21}$$

is *instance-optimal*: it achieves the best worst-case performance on any problem instance. It follows that the worst-case performance of any first-order method satisfying the span assumption can only be worse than that of (2.21). The worst-case performance of (2.21) being given by (2.20), we have that for any initialization $x_0 \in \mathbb{R}^d$ any first-order method satisfying the span assumption in Equation (2.8), there exists at least one problem on which

$$\|x_N - x_\star\|_2 \geq \max_{\substack{\mathbf{H}\in\mathcal{M},\, x_\star\in\mathbb{R}^d \\ \|x_0 - x_\star\|_2 \leq R}} \min_{\substack{P\in\mathcal{P}_N, \\ P(0)=1}} \|P(\mathbf{H})(x_0 - x_\star)\|_2, \tag{2.22}$$

where $x_N \in \mathbb{R}^d$ is the output of the first-order method under consideration, and $x_\star$ is the optimal point of the problem.

In other words, the max term in (2.22) searches for the "most difficult" quadratic function, while the min term represents the *best* first-order method for a specific quadratic function. Of course, the method (2.21) is much more powerful than the Chebyshev method, since it is optimal *for any specific function*. However, it is possible to show that despite being more powerful, this optimal algorithm has the same worst-case performance as that of Chebyshev's method when the dimension d of the ambient space is large enough. The lower bound result is summarized by the next theorem.

Theorem 2.3. (Nemirovsky, 1994, Proposition 12.3.2) Let $N, d \in \mathbb{N}$ such that $d \geq N+1$, and let $x_0 \in \mathbb{R}^d$. There exists $\mathbf{H} \in \mathbf{S}_d : \mu\mathbf{I} \preceq \mathbf{H} \preceq L\mathbf{I}$ and $x_\star \in \mathbb{R}^d$ such that any sequence $\{x_k\}_{k=0,\ldots,N}$ generated by any first-order method satisfying (2.8), and initiated at x_0 for minimizing the quadratic function f in the form (2.1) satisfies

$$\|x_N - x_\star\|_2 \geq \|x_0 - x_\star\|_2 \min_{\substack{P\in\mathcal{P}_N, \\ P(0)=1}} \max_{\lambda\in[\mu,\, L]} |P(\lambda)| = \frac{2}{\xi^N + \xi^{-N}} \|x_0 - x_\star\|_2,$$

with $\xi = \dfrac{\sqrt{\frac{L}{\mu}}+1}{\sqrt{\frac{L}{\mu}}-1}$.

More details on this topic can be found in (Nemirovsky, 1994, Section 12.3).

2.4 Notes and References

The Chebyshev method presented in this section is *worst-case* optimal for the class of quadratic functions with Hessians $\mathbf{H}$ satisfying $\mu\mathbf{I} \preceq \mathbf{H} \preceq L\mathbf{I}$. More detailed discussions and developments on the topic of Chebyshev polynomials, for quadratic minimization, are provided in (Nemirovsky and Yudin, 1983a; Nemirovsky, 1992; Nesterov, 2003), as well as in the lecture notes (Nemirovsky, 1994, Chapter 10). Those references include the treatment of the case where the smallest eigenvalue is $\mu = 0$. Finally, one should note that the optimal convergence bounds achieved by the Chebyshev method requires knowledge of the problem class parameters, μ and L, which might or might not be an issue, depending on the problem at hand.

Probably the most celebrated method for unconstrained quadratic optimization problems is the conjugate gradient (CG) method. Its origin is usually attributed to Stiefel (1952) and Straeter (1971). As for the setup of this section, it turns out that CG methods are *instance-optimal*, in the sense that they are the best performing first-order methods on every particular problem instance in the range of unconstrained quadratic minimization problems (in particular, the CG variant presented in (2.21) achieves the lower bound from Theorem 2.3). The classical CG produces iterates $\{x_k\}_{k\geq 0}$ such that

$$\begin{aligned} x_{k+1} \in \operatorname*{argmin}_x \; & f(x) \\ & \text{s.t. } x \in x_0 + \operatorname{span}\{H(x_0 - x_\star), \ldots, H^k(x_0 - x_\star)\}, \end{aligned}$$

which admits efficient formulation; see, e.g., (Nocedal and Wright, 2006). Another variant of CG is often referred to as MINRES, which produces iterates $\{x_k\}_{k\geq 0}$ in the form

$$\begin{aligned} x_{k+1} \in \operatorname*{argmin}_x \; & \|\nabla f(x)\|_2 \\ & \text{s.t. } x \in x_0 + \operatorname{span}\{H(x_0 - x_\star), \ldots, H^k(x_0 - x_\star)\}. \end{aligned}$$

Its generalization GMRES (Saad and Schultz, 1986) is popular for solving linear systems of the form $\mathbf{H}x = b$ when $\mathbf{H}$ is not required to be either symmetric or invertible.

Yet another alternative for dealing with quadratic minimization is to resort on *Anderson*-type acceleration schemes. As for conjugate gradient methods, those schemes do not readily extend beyond quadratic minimization with the same nice theoretical guarantees. This is the topic of the next section.

Beyond quadratic optimization, properties of Chebyshev polynomials is the focus of (Mason and Handscomb, 2002). The use of Chebyshev polynomials in the context of solving linear systems is covered at length in (Fischer, 1996). In particular, the theory of (Fischer, 1996) can be instantiated for the convex quadratic minimization in average-case analyses, where Chebyshev polynomials (along with their heavy-ball limits) also naturally appear (Pedregosa and Scieur, 2020; Lacotte and Pilanci, 2020; Scieur and Pedregosa, 2020).

3

Nonlinear Acceleration

In this section, we see that the main argument used in the Chebyshev method can be adapted beyond quadratic problems. The extension that we present here, called nonlinear acceleration, follows a pattern that is known in numerical analysis as *vector extrapolation methods*: it seeks to accelerate the convergence of sequences by extrapolation using nonlinear averages. Different such strategies are known under various names, starting with Aitken's Δ^2 (Aitken, 1927), Wynn's epsilon algorithm (Wynn, 1956), and Anderson acceleration (Anderson, 1965); a survey of these techniques can be found in (Sidi *et al.*, 1986). The vector extrapolation techniques, generic by nature, can be applied to optimization, as explained in what follows.

3.1 Introduction

This section focuses on the convex minimization problem:

$$\text{minimize } f(x) \tag{3.1}$$

in the variable $x \in \mathbb{R}^d$. We assume f to be twice continuously differentiable in a neighborhood of its minimizer $x_\star$, and denote by $f_\star = f(x_\star)$ the minimum of f.

We aim at adapting some of the ideas behind Chebyshev's acceleration (see Section 2) to a broader class of convex minimization problems beyond quadratic minimization. These adaptations stems from a local quadratic approximation of the objective:

$$f(x) = f_\star + \frac{1}{2}\langle x - x_\star; \mathbf{H}(x - x_\star)\rangle + o(\|x - x_\star\|_2^2), \tag{3.2}$$

where $\mathbf{H} = \nabla^2 f(x_\star) \in \mathbf{S}_d$ (the set of symmetric $d \times d$ matrices) is the Hessian of f at $x_\star$, which we assume to satisfy $\mu\mathbf{I} \preceq \mathbf{H} \preceq L\mathbf{I}$ for some $0 < \mu < L$. Of course, neglecting the second-order term in (3.2) allows recovering a quadratic minimization problem for which one could apply Chebyshev's method as is.

We recall that the Chebyshev method is the first-order method associated with the *best worst-case polynomial*. In short, the kth iteration of Chebyshev's method consists in combining previous gradients for minimizing a worst-case convergence bound over all μ-strongly convex L-smooth quadratic problems in $\mathbb{R}^d$ (in the form (2.1) and with $d \geq k+1$):

$$\begin{aligned} \alpha_k = \underset{\{\alpha^{(i)}\}_i}{\operatorname{argmin}} \max_{\substack{\mu \mathbf{I} \preceq \mathbf{H} \preceq L\mathbf{I} \\ x_0,\dots,x_k,\, x_\star,\, b \in \mathbb{R}^{k+1}}} & \frac{\|x_k - x_\star\|_2^2}{\|x_0 - x_\star\|_2^2} \\ & \text{s.t. } x_k = x_0 - \sum_{i=0}^{k-1} \alpha^{(i)} \nabla f(x_i), \\ & \mathbf{H} x_\star = b, \end{aligned} \tag{3.3}$$

$$x_k = x_0 - \sum_{i=0}^{k-1} \alpha_k^{(i)} \nabla f(x_i),$$

so that α_k does not depend on the particular problem instance $(\mathbf{H}, x_\star)$ and on the initialization x_0, but only on the problem class described by μ and L (we note that in Section 2, (3.3) was expressed in terms of optimizing a polynomial (2.12)). A natural alternative to Chebyshev's method consists in choosing those weights *adaptively*. That is, depending on the particular instance of the problem at hand. For doing so, we have to choose another way to measure performance (because minimizing $\|x_k - x_\star\|_2$ would require knowledge of $x_\star$); one such possibility is to rely on function values or gradient norms. One could then rely on conjugate gradient-type methods which are very attractive for unconstrained quadratic minimization (see, e.g., discussions in Section 2.4). In this section, we consider the case where a first-order optimization method provided us with a sequence of pairs $\{(x_i, \nabla f(x_i)\}_{i=0,\dots,k}$ satisfying (for $i = 1, \dots, k$)

$$x_i \in x_0 + \operatorname{span}\left\{\nabla f(x_0),\, \nabla f(x_1),\, \dots,\, \nabla f(x_{i-1})\right\}, \tag{3.4}$$

and we study methods producing approximations of $x_\star$ as linear combinations $\sum_{i=0}^k c_i x_i$ of the previous iterates $\{x_i\}_{i=0,\dots,k}$. For choosing the corresponding weights, we minimize the norm of the gradient at the approximated point. In unconstrained convex quadratic minimization problems, this approach is closely related to the so-called MINRES (Paige and Saunders, 1975) and GMRES (Saad and Schultz, 1986) methods (conjugate gradient-type methods minimizing gradient norms; see discussions by Walker and Ni (2011) and Section 2.4). That is, when f is quadratic, we choose the weights $\{c_i\}_{i=0,\dots,k}$ by solving

$$c_\star = \underset{c}{\operatorname{argmin}} \left\{ \left\| \nabla f\left(\textstyle\sum_{i=0}^k c_i x_i\right) \right\|_2^2 : \mathbf{1}^T c = 1 \right\}. \tag{3.5}$$

Whereas the new approximation is a linear combination of previous iterates $\{x_i\}_{i=0,\dots,k}$, the coefficients $\{c_i\}_{i=0,\dots,k}$ depend *nonlinearly* on both ∇f and on $\{x_i\}_{i=0,\dots,k}$. This technique is known under a few different names including those of *Anderson acceleration* and *minimal polynomial extrapolation* (see discussions and references in Section 3.6 for

more details). In this section, we refer to all these methods as "nonlinear acceleration" techniques.

3.2 Nonlinear Acceleration for Quadratic Minimization

In this section, we present the simplest form of nonlinear acceleration, which is often referred to as the *offline* nonlinear acceleration mechanism. We start with the main arguments underlying the technique, and present a few variants later in this section.

The core idea of the mechanism is to use a sequence of iterates $\{x_i\}_{i=0,\dots,k}$ provided by a first-order method for solving (3.1). On this basis, we generate a new approximation of a solution to (3.1) as a linear combination of past iterates, in the form $x_{\text{extr}} = \sum_{i=0}^{k} c_i x_i$. The point x_{extr} is commonly referred to as the *extrapolation* and can be chosen in different ways. In classical nonlinear acceleration mechanisms, it is chosen for making $\|\nabla f(x_{\text{extr}})\|_2$ small, as in (3.5). In general, solving (3.5) is just as costly as solving (3.1), but the mechanism turns out to have an efficient formulation when minimizing quadratic functions of the form

$$f(x) = \frac{1}{2}\langle x - x_\star; \mathbf{H}(x - x_\star)\rangle + f_\star. \tag{3.6}$$

In this case, it is possible to find an explicit formula for (3.5). Indeed, the gradient of f is then a linear function, and because the coefficients $\{c_i\}_{i=0,\dots,k}$ sum to one, the gradient of the linear combination is equal to a linear combination of gradients:

$$\nabla f\left(\sum_{i=0}^{k} c_i x_i\right) = \mathbf{H}\left(\sum_{i=0}^{k} c_i x_i - x_\star\right) = \sum_{i=0}^{k} c_i \mathbf{H}\left(x_i - x_\star\right) = \sum_{i=0}^{k} c_i \nabla f(x_i). \tag{3.7}$$

It follows that (3.5) reduces to a simple quadratic program involving gradients of the past iterates. It can be formulated as

$$c_\star = \operatorname*{argmin}_{c} \left\{ \left\| \textstyle\sum_{i=0}^{k} c_i \nabla f\left(x_i\right) \right\|_2^2 : \; c^T \mathbf{1} = 1 \right\}. \tag{3.8}$$

For convenience, we use the following more compact form in the sequel

$$c_\star = \operatorname*{argmin}_{c^T \mathbf{1} = 1} \|\mathbf{G} c\|_2^2, \tag{3.9}$$

where $\mathbf{G} = [\nabla f(x_0), \dots, \nabla f(x_k)]$ is the matrix formed by concatenating past gradients. This quadratic subproblem requires solving a small linear system of $(k+1)$ equations. When $\mathbf{G}^T\mathbf{G}$ is invertible, an explicit solution is provided by

$$c_\star = \frac{z}{\sum_{i=0}^{k} z_i}, \quad \text{where } z = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{1}.$$

This mechanism is summarized in Algorithm 3.

3.2.1 Worst-case Convergence Bounds

In this section, we quantify the accuracy of nonlinear acceleration. In particular, we show that it is instance-optimal and achieves the same worst-case convergence rate as

Algorithm 3 Nonlinear acceleration (offline version)

Input: Sequence of pairs $\{(x_i, \nabla f(x_i))\}_{i=0,\dots,k}$.

1: Form the matrix $\mathbf{G} = [\nabla f(x_0), \dots, \nabla f(x_k)]$, and compute $\mathbf{G}^T\mathbf{G}$.

2: Solve the linear system $(\mathbf{G}^T\mathbf{G})z = \mathbf{1}$, and set $c = \frac{z}{z^T\mathbf{1}}$.

3: Form the extrapolated point $x_{\text{extr}} = \sum_{i=0}^{k} c_i x_i$.

Output: Approximate solution x_{extr}.

that Chebyshev's method (see Theorem 2.1 and Theorem 2.3) in the worst-case, as soon as the sequence $\{x_i\}_{i=0,\dots,k}$ is generated by a reasonable first-order method. Before going into the analysis, we introduce a few technical ingredients specifying what is a *reasonable* first-order method. In short, we require that $\{x_i\}_{i=0,\dots,k}$ is obtained from a "nondegenerate" first-order method. That is, we assume that the method uses $\nabla f(x_i)$ non-trivially for generating x_{i+1} (for all $i = 0, \dots, k-1$).

Definition 3.1 (Nondegenerate first-order method)**.** Let $x_0 \in \mathbb{R}^d$ be an initial point, and $f : \mathbb{R}^d \to \mathbb{R}$ be a continuously differentiable convex function. A first-order method generates sequences of iterates $\{x_i\}_{i=0,1,\dots}$ such that for all $i = 1, 2, \dots$

$$x_i \in x_0 + \operatorname{span}\{\nabla f(x_0), \dots, \nabla f(x_{i-1})\}.$$

A first-order method is *nondegenerate* if for all continuously differentiable convex function $f : \mathbb{R}^d \to \mathbb{R}$, all $i = 0, 1, \dots$, and all $x_0 \in \mathbb{R}^d$ there exists some $\{\alpha_j^{(i)}\}_{j=0,\dots,i} \subset \mathbb{R}$ with $\alpha_i^{(i)} \neq 0$ such that

$$x_{i+1} = x_0 + \sum_{j=0}^{i} \alpha_j^{(i)} \nabla f(x_j). \tag{3.10}$$

The proposition below shows that when the sequence $\{x_i\}_{i=0,1,\dots}$ is generated by a nondegenerate first-order method, the gradient of any iterate x_i can be written using a polynomial of degree *exactly* i.

Proposition 3.1. Let $x_0 \in \mathbb{R}^d$ be an initial point, $f : \mathbb{R}^d \to \mathbb{R}$ be a quadratic function in the form (3.6) with $\mu\mathbf{I} \preceq \mathbf{H} \preceq L\mathbf{I}$, and let $\{x_i\}_{i=0,1,\dots}$ be generated by a nondegenerate first-order method. Then, for all $i = 0, 1, \dots$, there exists a polynomial P_i of degree *exactly* i such that $P_i(0) = 1$ and

$$\nabla f(x_i) = P_i(\mathbf{H}) \nabla f(x_0).$$

Proof. We proceed by induction. First, we have $P_0 = 1$ (P_0 has degree 0 and $P_0(0) = 1$) and hence

$$\nabla f(x_0) = P_0(\mathbf{H}) \nabla f(x_0),$$

thereby trivially reaching the desired conclusion for $i = 0$.

We proceed with the induction hypothesis, assuming that the desired result holds for x_i. By Definition 3.1 (nondegenerate first-order method), and because f is a quadratic

function (3.6), we have

$$\begin{aligned}\nabla f(x_{i+1}) =& \nabla f\left(x_0 + \sum_{j=0}^{i} \alpha_j^{(i)} \nabla f(x_j)\right) \\ =& \mathbf{H}\left(x_0 - x_\star + \sum_{j=0}^{i} \alpha_j^{(i)} \nabla f(x_j)\right) \\ =& \mathbf{H}(x_0 - x_\star) + \sum_{j=0}^{i} \alpha_j^{(i)} \mathbf{H} \nabla f(x_j) \\ =& \nabla f(x_0) + \sum_{j=0}^{i} \alpha_j^{(i)} \mathbf{H} \nabla f(x_j).\end{aligned}$$

Thanks to the induction hypothesis, we have $\nabla f(x_j) = P_j(\mathbf{H}) \nabla f(x_0)$ with $P_j(0) = 1$ and $\deg(P_j) = j$ for $j \leq i$. For showing that P_{i+1} satisfies the desired claim, we start by expressing $\nabla f(x_{i+1})$ in terms of a polynomial:

$$\nabla f(x_{i+1}) = \underbrace{\left(P_0(\mathbf{H}) + \sum_{j=0}^{i} \alpha_j^{(i)} \mathbf{H} P_j(\mathbf{H})\right)}_{= P_{i+1}(\mathbf{H})} \nabla f(x_0).$$

It is relatively straightforward to verify $P_{i+1}(0) = 1$ using the previous expression:

$$P_{i+1}(0) = P_0(0) + \sum_{j=0}^{i} \alpha_j^{(i)} \cdot 0 \cdot P_j(0) = P_0(0) = 1.$$

Finally, a minor reorganization of the expression of P_{i+1} allows writing

$$P_{i+1}(\mathbf{H}) = \underbrace{P_0(\mathbf{H}) + \sum_{j=0}^{i-1} \alpha_j^{(i)} \mathbf{H} P_j(\mathbf{H})}_{\text{degree} \leq i} + \alpha_i^{(i)} \mathbf{H} P_i(\mathbf{H}).$$

Nondegeneracy of the first-order method implies that there exists some $\alpha_i^{(i)} \neq 0$ such that the previous expression holds. Finally it follows from $\deg(P_i) = i$ (induction hypothesis) that $\deg(\mathbf{H} P_i(\mathbf{H})) = i + 1$, thereby reaching the desired claim. ∎

Equipped with previous technical ingredients, one can show that Algorithm 3 is "instance-optimal" when applied to a nondegenerate first-order method. This means that the nonlinear acceleration algorithm finds the *best polynomial* given a *specific* quadratic function f—in opposition to the Chebyshev method that finds the best polynomial for a *class* of functions (Theorem 2.1). In other words, nonlinear acceleration *adaptively* looks for the best combination of previous iterates given the information stored in the previous gradients, while Chebyshev's method uses the same worst-case optimal polynomial in all cases. Moreover, Algorithm 3 does *not* require knowledge of the smoothness or strong convexity parameters.

Theorem 3.1 (Instance-optimality of nonlinear acceleration). Let $x_0 \in \mathbb{R}^d$ be an initial point, f be the quadratic function from (3.6) with $\mu\mathbf{I} \preceq \mathbf{H} \preceq L\mathbf{I}$, and $\{x_i\}_{i=0,\dots,k}$ be generated by a nondegenerate first-order method initiated from x_0. For any $k \geq 0$, it holds that

$$\begin{aligned} \|\nabla f(x_{\text{extr}})\|_2 &= \min_{c^T\mathbf{1}=1} \left\|\nabla f\left(\textstyle\sum_{i=0}^k c_i x_i\right)\right\|_2 \\ &= \min_{\substack{P\in\mathcal{P}_k,\\ P(0)=1}} \|P(\mathbf{H})\nabla f(x_0)\|_2, \end{aligned} \tag{3.11}$$

where x_{extr} is obtained from Algorithm 3 $\{(x_i, \nabla f(x_i))\}_{i=0,\dots,k}$, and $\mathcal{P}_k$ is the set of polynomials of degree at most k.

Proof. Since the coefficients c_i sum to one, we have the following equalities:

$$\begin{aligned} \nabla f(x_{\text{extr}}) &= \nabla f\left(\sum_{i=0}^k c_i x_i\right) = \mathbf{H}\left(\sum_{i=0}^k c_i (x_i - x_\star)\right) \\ &= \left(\sum_{i=0}^k c_i \mathbf{H}(x_i - x_\star)\right) = \sum_{i=0}^k c_i \nabla f(x_i). \end{aligned}$$

Therefore,

$$\|\nabla f(x_{\text{extr}})\|_2 = \left\|\sum_{i=0}^k c_i \nabla f(x_i)\right\|_2.$$

We now use the definition of a first-order method, which yields iterates x_i such that

$$x_i \in x_0 + \text{span}\{\nabla f(x_0), \dots, \nabla f(x_{i-1})\}.$$

From Section 2.3, Proposition 2.1, it follows that x_i can be written as

$$x_i = x_\star + P_i(\mathbf{H})(x_0 - x_\star), \quad P_i \in \mathcal{P}_i, \quad P_i(0) = 1.$$

It also holds that its gradient can be written using the same polynomial:

$$\begin{aligned} \nabla f(x_i) &= \mathbf{H}(x_i - x_\star) = \mathbf{H}P_i(\mathbf{H})(x_0 - x_\star) = P_i(\mathbf{H})(\mathbf{H}(x_0 - x_\star)) \\ &= P_i(\mathbf{H})\nabla f(x_0). \end{aligned} \tag{3.12}$$

By substituting this expression in the objective of (3.8), we obtain

$$\min_{c^T\mathbf{1}=1} \left\|\sum_{i=0}^k c_i \nabla f(x_i)\right\|_2^2 = \min_{c^T\mathbf{1}=1} \left\|\left(\sum_{i=0}^k c_i P_i(\mathbf{H})\right)\nabla f(x_0)\right\|_2^2. \tag{3.13}$$

Since the iterates $\{x_i\}_{i=0,\dots,k}$ are generated by a nondegenerate first-order method, all polynomials P_i have differents degrees. Therefore, the polynomials $\{P_i\}_{i=0,\dots,k}$ are linearly independent, and hence $\{P_i\}_{i=0,\dots,k}$ is a basis for the space $\mathcal{P}_k$. Finally, because $P_i(0) = 1$ and $c^T\mathbf{1} = 1$, we can rephrase the objective (3.13) as

$$\min_{c^T\mathbf{1}=1} \left\|\left(\sum_{i=0}^k c_i P_i(\mathbf{H})\right)\nabla f(x_0)\right\|_2 = \min_{\substack{P\in\mathcal{P}_k,\\ P(0)=1}} \|P(\mathbf{H})\nabla f(x_0)\|_2.$$

The convergence rate is thus given by

$$\|\nabla f(x_{\text{extr}})\|_2 \;=\; \left\|\sum_{i=0}^{k} c_i \nabla f(x_i)\right\|_2 \;=\; \min_{\substack{P\in\mathcal{P}_k,\\ P(0)=1}} \|P(\mathbf{H})\nabla f(x_0)\|_2\,.$$

■

This theorem shows that the worst-case convergence rate is essentially controlled by the optimal value of a minimization problem. The minimum in (3.11) can be bounded using a Chebyshev argument similar to the main argument used in Section 2.

Corollary 3.2. Let $x_0 \in \mathbb{R}^d$ be an initial point, f be the quadratic function from (3.6) with $\mu\mathbf{I} \preceq \mathbf{H} \preceq L\mathbf{I}$, and $\{x_i\}_{i=0,\dots,k}$ be generated by a nondegenerate first-order method initiated from x_0. Then, for any $k \geq 0$, it holds that

$$\|\nabla f(x_{\text{extr}})\|_2 \leq \frac{2}{\xi^k + \xi^{-k}}\|\nabla f(x_0)\|_2, \quad \xi = \frac{\sqrt{\frac{L}{\mu}}+1}{\sqrt{\frac{L}{\mu}}-1}.$$

where x_{extr} is the output of Algorithm 3 applied to $\{(x_i,\, \nabla f(x_i))\}_{i=0,\dots,k}$.

Proof. We use the shifted Chebyshev polynomial from Theorem 2.1 as a feasible solution of the minimization problem (3.11). ■

As for the Chebyshev method, it is possible to show that the convergence rate cannot be improved as it matches that of the corresponding lower bound, which can be obtained by adapting Theorem 2.3 to gradient norms; see e.g., (Nemirovsky, 1994, Proposition 12.3.2).

Remark 3.1 (Finite-time convergence)**.** When $\{(x_i,\, \nabla f(x_i))\}_{i=0,\dots,k}$ is generated by a nondegenerate first-order method and when k is large enough, nonlinear acceleration eventually converges *exactly* to the minimizer of the quadratic function f (this follows easily from analogies with conjugate gradient-type methods; see, e.g.,Section 2.4). More formally, for all $k \geq d$ it holds that

$$x_{\text{extr}} = x_\star, \tag{3.14}$$

where x_{extr} is the output of Algorithm 3 applied to $\{(x_i,\, \nabla f(x_i))\}_{i=0,\dots,k}$. This is a natural consequence of the fact that x_{extr} is the best point in $x_0 + \text{span}\{\nabla f(x_0), \dots, \nabla f(x_{k-1})\}$, as provided by (3.8).

3.2.2 Computational Complexity

The computational complexity of Algorithm 3 is $O(dk^2 + k^3)$, where k is the length of the sequence of gradients and d is the dimension of the ambient space. The first term originates from the matrix-matrix multiplication $\mathbf{G}^T\mathbf{G}$ in step 1, and the second one comes from solving a $(k+1) \times (k+1)$ matrix in step 2. The length k being often much smaller than d, the resulting complexity is typically $O(dk^2)$.

Low-rank updates. When using the nonlinear acceleration method in parallel with a first-order method generating a growing sequence of iterates $\{x_i\}_{i=0,\dots,k}$, an extrapolation step can be computed each time a new iterate is produced. we can reduce the per iteration complexity up to $O(dk)$ by computing the matrix $\mathbf{G}^T\mathbf{G}$ and the coefficients $c_\star$ using low-rank updates (Sidi, 1991).

Limited-memory. Because the iteration complexity of nonlinear acceleration grows with the length of the input sequence, it might become costly to compute an extrapolation. It is therefore common to use nonlinear acceleration only with the last few iterates produced by the first-order method. More formally, if we impose a maximum memory of m pairs, we compute the extrapolation as follow when $k \geq m$. More formally, if we impose a maximum memory of m pairs, one can use Algorithm 3 with input $\{(x_{k-m+i}, \nabla f(x_{k-m+i}))\}_{i=1,\dots,m}$.

3.2.3 Online Nonlinear Acceleration

So far, we have seen a post-processing procedure that generates an extrapolated point x_{extr} from a sequence of pairs $\{x_i, \nabla f(x_i)\}_{i=0,\dots,k}$. If this sequence is generated by a nondegenerate first-order method, then Corollary 3.2 shows that the gradient of the extrapolated point x_{extr} converges to zero at an optimal worst-case convergence rate, without any hyper-parameters. Perhaps surprisingly, Algorithm 3 is itself not a nondegenerate first-order method, and can therefore not be used recursively as is.

In what follows, we introduce a *mixing* parameter. This parameter transforms the nonlinear acceleration mechanism to a nondegenerate first-order method without hurting its worst-case performance. This enables using nonlinear acceleration recursively for generating the whole sequence $\{x_i\}_{i=1,2,\dots}$. This technique is often referred to as *online nonlinear acceleration*.

Mixing parameter. The idea underlying the mixing parameter is fairly simple: instead of combining previous iterates, we combine *gradient steps* as follows:

$$x_{\text{extr}}^{\text{mixing}} = \sum_{i=0}^{k} c_i \Big(x_i - h \nabla f(x_i) \Big), \tag{3.15}$$

as provided by Algorithm 4. Furthermore, the use of an appropriate step size h can even slightly improve the worst-case convergence speed of Algorithm 3.

Algorithm 4 Nonlinear acceleration (with mixing)

Input: Sequence of pairs $\{(x_i, \nabla f(x_i))\}_{i=0,\dots,k}$, mixing parameter h.

1: Form the matrix $\mathbf{G} = [\nabla f(x_0), \dots, \nabla f(x_k)]$, and compute $\mathbf{G}^T\mathbf{G}$.
2: Solve the linear system $(\mathbf{G}^T\mathbf{G})z = \mathbf{1}$, and compute $c = \frac{z}{z^T\mathbf{1}}$.
3: Form the extrapolated point $x_{\text{extr}} = \sum_{i=0}^{k} c_i(x_i - h\nabla f(x_i))$.

Output: Approximate solution x_{extr}.

Intuitively, this *mixing* between iterates and gradients emulates a gradient step on the extrapolated point from Algorithm 3, that we call here $x_{\text{extr}}^{\text{offline}} = \sum_{i=0}^{k} c_i x_i$,

$$\begin{aligned} x_{\text{extr}}^{\text{mixing}} &= \sum_{i=0}^{k} c_i x_i - h \sum_{i=0}^{k} c_i \nabla f(x_i) \\ &= \sum_{i=0}^{k} c_i x_i - h \nabla f\left(\sum_{i=0}^{k} c_i x_i \right) \\ &= x_{\text{extr}}^{\text{offline}} - h \nabla f(x_{\text{extr}}^{\text{offline}}), \end{aligned}$$

where the second equality comes from the fact that f is a quadratic function, and that $\sum_{i=0}^{k} c_i = 1$.

This mixing parameter requires tuning one hyper-parameter h, which can be chosen in various ways. Proposition 3.2 shows that the mixing strategy slightly improves the performance of nonlinear acceleration if h is set properly.

Proposition 3.2. Let $x_0 \in \mathbb{R}^d$ be an initial point, f be the quadratic function from (3.6) with $\mu \mathbf{I} \preceq \mathbf{H} \preceq L\mathbf{I}$, and $\{x_i\}_{i=0,\dots,k}$ be generated by a nondegenerate first-order method initiated from x_0. Then, for any $k \geq 0$, it holds that

$$\|\nabla f(x_{\text{extr}}^{\text{mixing}})\|_2 \leq C_h \|\nabla f(x_{\text{extr}}^{\text{offline}})\|_2,$$

where $x_{\text{extr}}^{\text{offline}}$ is obtained from Algorithm 3 applied to $\{(x_i,\, \nabla f(x_i))\}_{i=0,\dots,k}$, $x_{\text{extr}}^{\text{mixing}}$ is the extrapolation with mixing from (3.15), and

$$C_h = \max\left\{1 - \mu h \;;\; Lh - 1\right\}.$$

Moreover, the factor C_h is guaranteed to be smaller than one if $h \in (0,\, \frac{2}{L})$, and takes its minimal value at $h = \frac{2}{L+\mu}$.

Proof. We start with the identity

$$x_{\text{extr}}^{\text{mixing}} = x_{\text{extr}}^{\text{offline}} - h \nabla f(x_{\text{extr}}^{\text{offline}}).$$

Because f is the quadratic function (3.6), the gradient of $x_{\text{extr}}^{\text{mixing}}$ reads

$$\begin{aligned} \nabla f(x_{\text{extr}}^{\text{mixing}}) &= \mathbf{H}(x_{\text{extr}}^{\text{mixing}} - x_\star) \\ &= \mathbf{H}(x_{\text{extr}}^{\text{offline}} - h \nabla f(x_{\text{extr}}^{\text{offline}}) - x_\star) \\ &= \nabla f(x_{\text{extr}}^{\text{offline}}) - \mathbf{H} h \nabla f(x_{\text{extr}}^{\text{offline}}) \\ &= (I - \mathbf{H}h) \nabla f(x_{\text{extr}}^{\text{offline}}). \end{aligned}$$

Therefore, we have the bound

$$\|\nabla f(x_{\text{extr}}^{\text{mixing}})\|_2 \leq \|I - \mathbf{H}h\|_2 \|\nabla f(x_{\text{extr}}^{\text{offline}})\|_2,$$

and the desired result follows from $\mu \mathbf{I} \preceq \mathbf{H} \preceq L\mathbf{I}$. ∎

Online nonlinear acceleration method. As previously underlined, the mixing parameter transforms the nonlinear acceleration method into a nondegenerate first-order method. We can therefore use it recursively. The online variant of the nonlinear acceleration technique, with limited memory, is provided in Algorithm 5, using Algorithm 4 as a subroutine. One should note that when $m = \infty$ (no memory restriction), the worst-case performance of offline version of nonlinear acceleration with mixing (Algorithm 4) is also valid for its online variant (Algorithm 5). It follows from Proposition 3.2 that the worst-case performance of Algorithm 4 and Algorithm 5 is no worse than that of offline version of nonlinear acceleration (Algorithm 3), provided by Corollary 3.2.

Algorithm 5 Nonlinear acceleration (online version, limited memory)

Input: A differentiable function f, initial point x_0, mixing parameter h, maximum memory parameter m (optional, $m = \infty$ by default).

Initialize Empty sequence $\mathcal{S}$ of pairs iterate/gradient.
for $k = 0, \ldots$ **do**
  Compute $\nabla f(x_k)$; append the pair $\mathcal{S} \leftarrow \mathcal{S} \cup \{(x_k, \nabla f(x_k))\}$.
  if $k \geq m$ **then**
    Discard the oldest pair from $\mathcal{S}$.
  end if
  Compute the extrapolation $x_{k+1} = [\text{Algorithm 4}](\mathcal{S}, h)$.
end for

Output: Approximate solution x_{k+1}.

In the next section, we see that nonlinear acceleration technique might suffer from serious instability issues when applied beyond quadratic minimization. Perhaps luckily, a simple regularization technique allows stabilizing the procedure beyond quadratics.

3.3 Regularized Nonlinear Acceleration Beyond Quadratics

Nonlinear acceleration suffers some serious drawbacks when used outside the restricted setting of quadratic functions. In fact, Algorithm 3 and Algorithm 4 are numerically highly unstable. This problem originates from the conditioning of the matrix $\mathbf{G}^T\mathbf{G}$, used to compute the coefficients $\{c_i\}_{i=0,\ldots,k}$. To illustrate this statement, assume we run Algorithm 3 with a noisy sequence of gradients $\{(x_i, \nabla f(x_i) + e_i)\}_{i=0,\ldots,k}$ where the sequence $\{e_i\}_{i=0,\ldots,k}$ is such that $\|e_i\|_2 \leq \epsilon$ for some $\epsilon > 0$. Scieur *et al.* (2016, Proposition 3.1) show that the relative distance between $\tilde{c}$ (the coefficients computed using the noisy sequence above) and its noise-free version c satisfies

$$\frac{\|c - \tilde{c}\|_2}{\|c\|_2} = O\left(\|\mathbf{E}\|_2 \ \|((\mathbf{G} + \mathbf{E})^T(\mathbf{G} + \mathbf{E}))^{-1}\|_2\right). \tag{3.16}$$

where $\mathbf{E} \triangleq [e_0, \ldots, e_k]$ is the noise matrix. In this bound, the perturbation impacts the solution proportionally to its norm and to the conditioning of $\mathbf{G} + \mathbf{E}$.

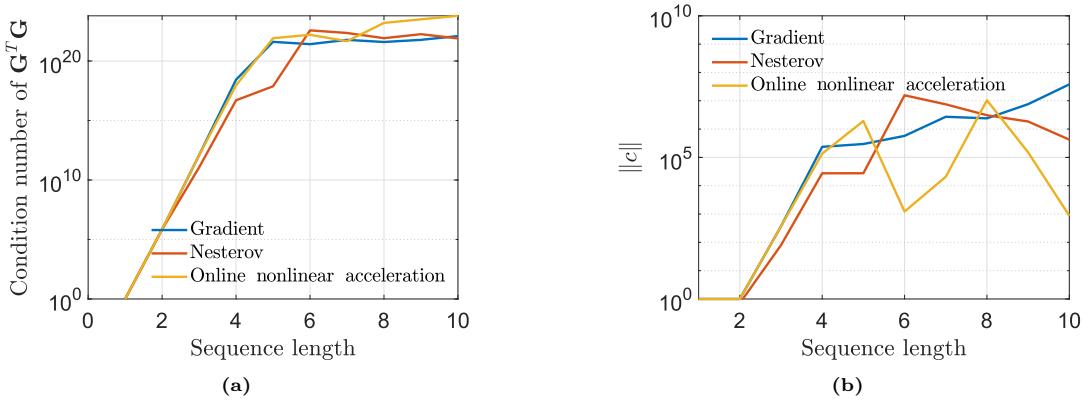

Figure 3.1: Illustration of the sensitivity of nonlinear acceleration when applying nonlinear acceleration to gradient descent, to Nesterov's method (see Section 4), and using its online variant (Algorithm 5) to minimize some random quadratic function. Figure 3.1a: the condition number of the matrix $\mathbf{G}^T\mathbf{G}$, which grows exponentially with its size (the plateau on the right is caused by numerical errors). Figure 3.1b: the norm of the vector of coefficients c.

Unfortunately, even for small perturbations, the condition number of $\mathbf{G} + \mathbf{E}$ and the norm of the vector c are usually *huge*. In fact, $\mathbf{G}$ has a *Krylov* matrix structure, which is notoriously poorly conditioned (Tyrtyshnikov, 1994). Thereby, even a very small perturbation $\mathbf{E}$ might have a significant impact on performance. For illustrating this, let us briefly illustrate the link between $\mathbf{G}$ and Krylov matrices: consider using gradient descent with step size γ on a quadratic function; the iterates follow the rule

$$x_{k+1} - x_\star = (1 - \gamma\mathbf{H})^k(x_0 - x_\star) \quad \Leftrightarrow \quad \nabla f(x_{k+1}) = (1 - \gamma\mathbf{H})^k \nabla f(x_0).$$

Thereby, a matrix $\mathbf{G}$ formed by these expressions has the form

$$\mathbf{G} = \Big[\nabla f(x_0),\, (\mathbf{I} - \gamma\mathbf{H})\nabla f(x_0),\, (\mathbf{I} - \gamma\mathbf{H})^2\nabla f(x_0),\, \ldots\Big],$$

which shows that $\mathbf{G}$ is in fact a *Krylov* matrix—by definition, a Krylov matrix K associated with a matrix A and vector v is defined as $K = [v,\, Av,\, A^2v, \ldots]$.

In Figure 3.1, we show the norm of c and the condition number of the matrix $\mathbf{G}^T\mathbf{G}$ when it is formed from iterates of gradient descent, accelerated gradient descent (see Section 4), and nonlinear acceleration (in the online setting, see Algorithm 5) for minimizing some randomly generated quadratic function. Figure 3.1a shows that even after 3 iterations, the system can already be considered singular (i.e., the condition number exceeds 10^{16}).

For stabilizing the method, it is common to regularize the linear system. The resulting algorithm is often referred to as regularized nonlinear acceleration (RNA) (Scieur *et al.*, 2016). The following section is devoted to some theoretical properties of this method.

3.3.1 Regularized Nonlinear Acceleration

Regularized nonlinear acceleration (RNA) consists of using Algorithm 3 with a regularization, thereby rendering the method less sensitive to noise. In short, the base operation

underlying RNA is to solve

$$\underset{c^T\mathbf{1}=1}{\operatorname{argmin}} \frac{\|\mathbf{G}c\|_2^2}{\|\mathbf{G}\|_2^2} + \lambda\|c - c_{\text{ref}}\|_2^2 \tag{3.17}$$

in the variable $c \in \mathbb{R}^k$, where c_{ref} is some reference vector. The effect of regularization is therefore to force c to be close to c_{ref}. Of course, it makes more sense to pick c_{ref} summing to one. A common choice is to pick $c_{\text{ref}} = \mathbf{1}/k$, which would enforce the procedure to be "not too far" from a simple averaging of the iterates. Another possibility is $c_{\text{ref}} = [\mathbf{0}_{k-1}, 1]$, which puts more weight on the last iterate. The division by $\|\mathbf{G}\|_2^2$ is for scaling purposes, as it makes λ dimensionless. The resulting method is slightly more complicated than its previous version without regularization and is provided in Algorithm 6. When $c_{\text{ref}} = \mathbf{1}/k$, the procedure simplifies to Algorithm 7.

Algorithm 6 Regularized nonlinear acceleration (RNA)

Input: Sequence of pairs $\{(x_i, \nabla f(x_i))\}_{i=0,\dots,k}$, mixing parameter h, regularization term $\lambda > 0$, reference vector c_{ref}.

1: Form $\mathbf{G} = [\nabla f(x_0), \dots, \nabla f(x_k)]$, compute $\mathcal{G} = \frac{\mathbf{G}^T\mathbf{G}}{\|\mathbf{G}^T\mathbf{G}\|_2}$.
2: Solve the linear system $(\mathcal{G} + \lambda\mathbf{I})w = \lambda c_{\text{ref}}$.
3: Solve the linear system $(\mathcal{G} + \lambda\mathbf{I})z = \mathbf{1}$.
4: Compute the coefficients $c = w + z\frac{(1-w^T\mathbf{1})}{z^T\mathbf{1}}$
5: Form the extrapolated point $x_{\text{extr}} = \sum_{i=0}^k c_i\,(x_i - h\nabla f(x_i))$.

Output: Approximate solution x_{extr}.

Algorithm 7 Regularized nonlinear acceleration (with $c_{\text{ref}} = \mathbf{1}/k$)

Input: Sequence of pairs $\{(x_i, \nabla f(x_i))\}_{i=0,\dots,k}$, mixing parameter h, regularization term $\lambda > 0$.

1: Form $\mathbf{G} = [\nabla f(x_0), \dots, \nabla f(x_k)]$ and compute $\mathcal{G} = \frac{\mathbf{G}^T\mathbf{G}}{\|\mathbf{G}^T\mathbf{G}\|_2}$.
2: Solve the linear system $(\mathcal{G} + \lambda\mathbf{I})z = \mathbf{1}/k$.
3: Compute the coefficients $c = \frac{z}{z^T\mathbf{1}}$.
4: Form the extrapolated point $x_{\text{extr}} = \sum_{i=0}^k c_i\,(x_i - h\nabla f(x_i))$.

Output: Approximate solution x_{extr}.

Online regularized nonlinear acceleration. As in the quadratic case, Algorithm 6 and Algorithm 7 could be used as subroutines in the online nonlinear acceleration method (Algorithm 5), thereby forming the regularized version of the online acceleration algorithm.

3.3.2 Perturbed Linear Gradients

In this section, we consider the problem of minimizing a twice continuously differentiable convex function f, as in (3.1), beyond quadratic problems. For doing that, we introduce

perturbed linear gradients. As before, we consider iterates $\{x_i\}_{i=0,1,\ldots}$ originating from a first-order method satisfying

$$x_{k+1} \quad \in \quad x_0 + \operatorname{span}\{\nabla f(x_0),\, \nabla f(x_1),\, \ldots,\, \nabla f(x_k)\}.$$

However, $\nabla f(x)$ is now no longer the gradient of a quadratic function. Instead, gradients of f can be written as a sum of the gradients of a quadratic function with a perturbation term $e(x)$, as follows:

$$\nabla f(x) = \mathbf{H}(x - x_\star) + e(x), \quad \text{with } \mathbf{H} : \mathbf{0} \prec \mu\mathbf{I} \preceq \mathbf{H} \preceq L\mathbf{I}. \tag{3.18}$$

Indeed, it follows from twice continuous differentiability of f that

$$f(x) = f_\star + \underbrace{\langle \nabla f(x_\star)}_{=0}; x - x_\star\rangle + \frac{1}{2}\langle x - x_\star; \nabla^2 f(x_\star)(x - x_\star)\rangle + O(\|x - x_\star\|_2^3).$$

Therefore, $f(x)$ can be approximated by the quadratic function (3.18) with $\mathbf{H} = \nabla^2 f(x_\star)$. Similarly, its gradient reads

$$\nabla f(x) = \underbrace{\nabla f(x_\star)}_{=0} + \nabla^2 f(x_\star)(x - x_\star) + e(x) = \mathbf{H}(x - x_\star) + e(x),$$

where $e(x)$ is the first-order Taylor remainder of the gradient. Thus, minimizing a non-quadratic function is equivalent to minimizing a perturbed quadratic one with a second-order error on its gradient:

$$e(x_k) = \nabla f(x_k) - \mathbf{H}(x_k - x_\star) \quad \left(\Rightarrow \quad \|e(x_k)\|_2 = O(\|x_k - x_\star\|_2^2)\right), \tag{3.19}$$

where $\mathbf{H} = \nabla^2 f(x_\star)$.

3.3.3 Convergence Bound

Using a perturbation argument, it is possible to derive a convergence guarantee for RNA. We state here a simplified version of (Scieur *et al.*, 2018, Theorem 3.2), which describes how regularization balances acceleration and stability in Algorithm 7. We discuss convergence rates in greater detail in what follows.

Theorem 3.3. Let $x_0 \in \mathbb{R}^d$, f be a twice continuously differentiable function with minimum $x_\star$ and whose gradient ∇f follows (3.18) with $\|e(x_i)\|_2 \leq \epsilon$. Let $\{x_i\}_{i=0,\ldots,k}$ be generated by a nondegenerate first-order method initiated from x_0. Then, for any $k \geq 0$, it holds that

$$\begin{aligned}
&\|\mathbf{H}(x_{\text{extr}} - x_\star)\|_2 \\
&\leq \|\mathbf{I} - h\mathbf{H}\|_2 \left(\underbrace{V_k^{[\mu,L]}(\lambda)\|\mathbf{H}(x_0 - x_\star)\|_2}_{\textbf{acceleration}} + \underbrace{O\left(\sqrt{1 + \frac{1}{\lambda}}\epsilon\right)}_{\textbf{stability}} \right),
\end{aligned}$$

where x_{extr} is the output of Algorithm 7 applied to $\{(x_i, \nabla f(x_i))\}_{i=0,\dots,k}$ with parameters h, $\lambda > 0$, and $V_k^{[\mu,L]}(\lambda)$ is a constant that corresponds to the maximum value on the interval $[\mu, L]$ of the *regularized Chebyshev polynomial*, i.e,

$$\begin{aligned} V_k^{[\mu,L]}(\lambda) &= \max_{x\in[\mu,L]} |C_k^{[\mu,L],\lambda}(x)|, \\ \text{with } C_k^{[\mu,L],\lambda} &= \operatorname*{argmin}_{\substack{P\in\mathcal{P}_k,\\ P(0)=1}} \max_{x\in[\mu,L]} P^2(x) + \lambda\|\mathbf{G}^T\mathbf{G}\|_2\|P\|_2^2, \end{aligned} \tag{3.20}$$

where $\|P\|_2$ is the norm of the vector of coefficients of the polynomial P.

This theorem states that regularization helps stabilizing the algorithm while slowing down the convergence rate. The regularized Chebyshev polynomial is somehow a mid-point between the classical shifted Chebyshev polynomial $C_k^{[\mu,L]}$ (from (2.15)) and the polynomial whose coefficients are defined by $\mathbf{1}/k$ (the polynomial that averages the iterates $\{x_i\}_{i=0,\dots,k}$). By construction, its maximum value is always larger than that of the Chebyshev polynomial, but the norm of its coefficients is smaller. Unfortunately, there is as yet no known explicit expression of the regularized Chebyshev polynomial. To the best of our knowledge, its value can nevertheless be computed numerically (Barré *et al.*, 2020b).

This mid-point between Chebyshev coefficients and the simple averaging of iterates is also natural in the context of noisy iterations. When the noise is negligible, a small regularization parameter combines the iterates $\{x_i\}_{i=0,\dots,k}$ using nearly the classical Chebyshev weights. When the noise is more substantial, a larger regularization parameter brings the vector of coefficients c closer to the average $\mathbf{1}/k$, thereby improving the "stability term" while rendering the "acceleration" less effective.

3.3.4 Asymptotic Convergence Rate

We briefly discuss the behavior of RNA when the initial point x_0 approaches the solution $x_\star$. In particular, the next proposition shows that if the perturbation magnitude ϵ decreases faster than $\|\mathbf{H}(x_0 - x_\star)\|_2$, the parameter λ can be adjusted to ensure an asymptotic convergence rate comparable to that of the Chebyshev method on quadratic problems (see Section 2).

Informally, the theorem exploits the fact that as x_0 approaches $x_\star$, f gets closer to its quadratic approximation around $x_\star$. Thereby, an appropriate tuning of RNA allows matching (asymptotically) the convergence rate of nonlinear acceleration on quadratics (see Theorem 3.1).

Proposition 3.3. Let $x_0 \in \mathbb{R}^d$, f be a twice continuously differentiable function with minimum $x_\star$, whose gradient ∇f follows (3.18) with $\|e(x_i)\|_2 \leq \epsilon$. Let $\{x_i\}_{i=0,\dots,k}$ be generated by a nondegenerate first-order method initiated from x_0. If we have

$$\epsilon = O(\|x_0 - x_\star\|_2^\alpha), \quad \alpha > 1,$$

and if we set $\lambda \propto \|x_0 - x_\star\|_2^s$ (proportional to $\|x_0 - x_\star\|_2^s$), where $0 < s < 2(\alpha - 1)$, then it holds that

$$\lim_{x_0 \to x_\star} \frac{\|\mathbf{H}(x_{\text{extr}} - x_\star)\|_2}{\|\mathbf{H}(x_0 - x_\star)\|_2} \leq \|\mathbf{I} - h\mathbf{H}\|_2 \frac{2}{\xi^k + \xi^{-k}} \quad \text{where} \quad \xi = \frac{\sqrt{\frac{L}{\mu}} + 1}{\sqrt{\frac{L}{\mu}} - 1},$$

where x_{extr} is the output of Algorithm 7 applied to the sequence $\{(x_i, \nabla f(x_i))\}_{i=0,\dots,k}$ with parameters h, $\lambda > 0$.

Proof. To simplify the notation, set $R \triangleq \|\mathbf{H}(x_0 - x_\star)\|_2$. We start from the result of Theorem 3.3 and divide both sides by R:

$$\frac{\|\mathbf{H}(x_{\text{extr}} - x_\star)\|_2}{R} \leq \|\mathbf{I} - h\mathbf{H}\|_2 \left(V_k^{[\mu,L]}(\lambda) + O\left(\sqrt{1 + \frac{1}{\lambda}} \frac{\epsilon}{R} \right) \right).$$

Since $\lambda \propto R^s$ and $\epsilon = O(R^\alpha)$,

$$\begin{aligned} &\frac{\|\mathbf{H}(x_{\text{extr}} - x_\star)\|_2}{R} \\ &\quad \leq \|\mathbf{I} - h\mathbf{H}\|_2 \left(V_k^{[\mu,L]}(\lambda) + \sqrt{O(R^{2(\alpha-1)}) + O(R^{2(\alpha-1)-s})} \right). \end{aligned}$$

When $x_0 \to x_\star$, we have $R \to 0$ and

$$\begin{aligned} R^{2(\alpha-1)} &\to 0 && \text{(since } \alpha > 1\text{)}, \\ R^{2(\alpha-1)-s} &\to 0 && \text{(since } s < 2(\alpha - 1)\text{)}. \end{aligned}$$

Finally, in (3.20) the regularization parameter $\lambda \|\mathbf{G}^T\mathbf{G}\|_2 = O(R^2) \to 0$. Since the (non-regularized) shifted Chebyshev polynomial $C_k^{[\mu,L]} = C_k^{[\mu,L],0}$ is a feasible solution of (3.20), we have the following bounds:

$$\begin{aligned} \left(V_k^{[\mu,L]}(\lambda)\right)^2 &= \max_{x \in [\mu,L]} |C_k^{[\mu,L],\lambda}(x)|^2, \\ &\leq \left\{ \max_{x \in [\mu,L]} |C_k^{[\mu,L],\lambda}(x)|^2 \right\} + \lambda \|\mathbf{G}^T\mathbf{G}\|_2 \|C_k^{[\mu,L],\lambda}\|_2^2, \\ &= \min_{\substack{P \in \mathcal{P}_k, \\ P(0)=1}} \max_{x \in [\mu,L]} P^2(x) + \lambda \|\mathbf{G}^T\mathbf{G}\|_2 \|P\|_2^2, \\ &\leq \left\{ \max_{x \in [\mu,L]} |C_k^{[\mu,L]}(x)|^2 \right\} + \lambda \|\mathbf{G}^T\mathbf{G}\|_2 \|C_k^{[\mu,L]}\|_2^2. \end{aligned}$$

As $\lambda \to 0$ we have that the upper bound on $(V_k^{[\mu,L]}(\lambda)$ converges to the maximum value of the regular (shifted) Chebyshev polynomial, thereby reaching the desired claim. ∎

In short, the previous theorem states that the asymptotic convergence rate matches the rate of Chebyshev's method as soon as $\lambda \propto \|x_0 - x_\star\|_2^s$ is decreasing (condition $s > 0$), but not too quickly compared to the perturbation magnitude $\epsilon = O(\|x_0 - x_\star\|_2^\alpha)$ (condition $s < 2(\alpha - 1)$), which is achievable only when $\alpha > 1$. This condition is met, for instance, when accelerating twice continuously differentiable functions with gradient descent: the error decreases as $O(\|x_0 - x_\star\|_2^2)$, see (3.19), and therefore $\alpha = 2 > 1$.

3.4 Extensions

The previous sections presented the nonlinear acceleration mechanism for unconstrained convex quadratic minimization. It also contained an analysis of its regularized version when applied beyond quadratics. In this section, we briefly cover two natural extensions: (i) the application of nonlinear acceleration to iterates that are corrupted by a stochastic noise, and (ii) the application of nonlinear acceleration to constrained/composite convex optimization problems when a projection/proximal operator is used.

Stochastic gradients. In the common situation where the first-order method under consideration only has access to stochastic estimates $\tilde{\nabla} f(x)$ of the gradient (satisfying $\mathbb{E}[\tilde{\nabla} f(x_i)] = \nabla f(x_i)$) one can adapt the perturbation model

$$\tilde{\nabla} f(x_i) = \mathbf{H}(x_i - x_\star) + e_i,$$

to e_i being the sum of a stochastic noise with a Taylor remainder. This is typically the case when applying RNA to stochastic gradient descent (SGD) and related methods. In this case, Theorem 3.3 holds in expectation under standard assumptions (Scieur *et al.*, 2017a), such as a bounded variance of e_i. However, the asymptotic convergence result from Proposition 3.3 may not be achieved. Indeed, in this setting, Proposition 3.3 also holds in expectation under the condition

$$\mathbb{E}[\|e(x_i)\|_2] = O(\|x_0 - x_\star\|_2^\alpha), \quad \alpha > 1.$$

Unfortunately, an asymptotic acceleration is not always possible. For instance, when trying to accelerate the fixed step SGD, we have $\epsilon = O(1)$ (i.e., $\alpha = 0$) and the asymptotic guarantee does not apply. This is probably not a surprise as this SGD does not converge to the optimum, hence there is no apparent reason for any sequence extrapolation technique to work at all. Fortunately, Algorithm 6 does usually work for "variance reduced" first-order methods (Scieur *et al.*, 2017a), such as SAG (Schmidt *et al.*, 2017), SAGA (Defazio *et al.*, 2014a), or SVRG (Johnson and Zhang, 2013).

Nonlinear acceleration of the proximal gradient method. It is common to apply first-order methods to composite convex minimization problems of the form:

$$\min_{x \in \mathbb{R}^d} \{F(x) \triangleq f(x) + h(x)\}, \tag{3.21}$$

where f is a smooth strongly convex function (this class of functions is used intensively in Section 4; see Definition 4.1) and h is a closed, proper, and convex function (i.e., h has a closed, non-empty, and convex epigraph) and whose *proximal operator* is available:

$$\operatorname{prox}_{\gamma h}(x) \triangleq \operatorname*{argmin}_z \left\{ \gamma h(z) + \frac{1}{2}\|x - z\|_2^2 \right\} \tag{3.22}$$

for some step size $\gamma > 0$. Problem (3.21) can then be approached iteratively via the proximal gradient method:

$$x_{k+1} = \operatorname{prox}_{\gamma h}(x_k - \gamma \nabla f(x_k)). \tag{3.23}$$

We omit most of the details on proximal algorithms; see Section 4 and Section 5 for more details and references. For instance, when h is the indicator function of a non-empty closed convex set $\mathcal{C}$, the proximal operator corresponds to an orthogonal projection onto $\mathcal{C}$ and the proximal gradient method reduces to the projected gradient method.

Unfortunately, a naive use of nonlinear acceleration on the iterates $\{x_i\}_{i=0,\dots,k}$ does not immediately work in this context, for several reasons. In particular, it is not possible to ensure that the extrapolated point x_{extr} belongs to $\mathbf{dom}(h)$ (or to the set $\mathcal{C}$ when h is an indicator function for $\mathcal{C}$). Moreover, due to the use of the proximal operator, the iterates $\{x_i\}_{i=0,\dots,k}$ do not necessarily satisfy the span assumption (3.4).

Recently, Mai and Johansson (2020) adapted the Anderson Acceleration method to handle a large class of constrained and non-smooth composite problems. The main idea is as follows: instead of accelerating the sequence $\{x_i\}_{i=0,\dots,k}$ generated by the proximal gradient method (3.23), we accelerate an alternate sequence $\{z_i\}_{i=0,\dots,k}$ which satisfies

$$z_{i+1} = \text{prox}_{\gamma h}(z_i) - \gamma \nabla f\left(\text{prox}_{\gamma h}(z_i)\right), \quad z_1 = x_0 - \gamma \nabla f(x_0).$$

This sequence $\{z_i\}_{i=0,\dots,k}$ corresponds to the sequence generated by (3.23) with the ordering of the gradient and proximal steps being swapped.

This trick allows obtaining convergence bounds for nonlinear acceleration in the proximal setup under very few changes in the algorithm. In particular, Mai and Johansson (2020) show that using Algorithm 8 in the presence of a proximal operator does not change the convergence analysis—using Clarke's generalized Jacobian (Clarke, 1990), semi-smoothness (Mifflin, 1977; Qi and Sun, 1993) and assuming that the function h is twice *epi-differentiable* and that h is twice-differentiable around the solution $x_\star$. We refer the reader to (Rockafellar and Wets, 2009, Section 13) for a comprehensive treatment of epi-differentiability.

Algorithm 8 Online regularized nonlinear acceleration with a proximal operator

Input: Differentiable function f, closed proper convex function h with proximal operator available, initial point x_0, step size γ, regularization term λ and reference vector c_{ref}.

1: **Initialize** $z_1 = x_0 - \gamma \nabla f(x_0)$, $x_1 = \text{prox}_{\gamma h}(z_1)$, empty sequence S of pairs iterate/gradient.
2: **for** $k = 1 \dots$ **do**
3: Compute $g_k = \frac{\gamma \nabla f(x_k) + z_k - x_k}{\gamma}$, then append the pair $S \leftarrow S \cup \{(z_k, g_k)\}$.
4: Compute the extrapolation $z_{k+1} = [\text{Algorithm 6}](\mathcal{S}, \gamma, \lambda, c_{\text{ref}})$.
5: $x_{k+1} = \text{prox}_{\gamma h}(z_{k+1})$.
6: **end for**

Output: Approximate solution x_{k+1}.

3.5 Globalization Strategies and Speeding-up Heuristics

As for many standard optimization methods, such as quasi-Newton methods, RNA only has local convergence guarantees beyond quadratics. Therefore, it is common to embed the mechanism with some globalization strategies, a.k.a. safeguards. Those strategies ensure not to deteriorate too much the performance of the initial first-order method in cases where RNA is used beyond its guaranteed range of applications. Those strategies can also be seen as speeding-up heuristics.

Descent condition. It is in general not guaranteed that the extrapolated point x_{extr} is better than any iterate of the sequence $\{x_i\}_{i=0,\dots,k}$ produced by the original method. This situation might for example occur when extrapolating with a bad mixing or regularization parameter, or simply when the error terms are too large. One classical way of limiting the impact of such problems is by checking some descent condition. For instance, one might consider "accepting" x_{extr} only if it is better than previous iterates $\{x_i\}_{i=0,\dots,k}$:

$$f\left(x_{\text{extr}}\right) < \min_{i\in\{0,\dots,k\}} f(x_i),$$

and to discard it otherwise.

Line-search. Nonlinear acceleration requires the selection of a mixing parameter, which might be difficult to tune in practice. One common trick is to choose it via a line-search strategy. That is, defining:

$$x_{\text{extr}}(h) = \sum_{i=0}^{k} \left(c_i x_i - h\nabla f(x_i)\right),$$

one can choose h by approximately solving $\operatorname{argmin}_h f\left(x_{\text{extr}}(h)\right)$.

3.6 Notes and References

Nonlinear acceleration techniques have been studied extensively during recent decades, and excellent reviews can be found in (Smith *et al.*, 1987; Jbilou and Sadok, 1991; Brezinski and Zaglia, 1991; Jbilou and Sadok, 1995; Jbilou and Sadok, 2000; Brezinski, 2001; Brezinski and Redivo–Zaglia, 2019). The first usage of an acceleration technique for fixed point iteration can be traced back to (Gekeler, 1972; Brezinski, 1971; Brezinski, 1970).

There are numerous independent works leading to methods similar to those described here. The most classical, and probably the most similar, is *Anderson acceleration* (Anderson, 1965), which corresponds exactly to the online mode of nonlinear acceleration (without regularization). Despite it being an old algorithm, there has been a recent uptake of interest in the convergence analysis (Walker and Ni, 2011; Toth and Kelley, 2015) of Anderson acceleration thanks to its good empirical performance, and strong connection with quasi-Newton methods (Fang and Saad, 2009).

Other versions of nonlinear acceleration use different arguments but behave similarly. For instance, minimal polynomial extrapolation (MPE), which uses the properties of the minimal polynomial of a matrix (Cabay and Jackson, 1976); reduced rank extrapolation (RRE); and the Mesina method (Mešina, 1977; Eddy, 1979) are also variants of Anderson acceleration. The properties and equivalences of these approaches have been studied extensively during the past decades (Sidi, 1988; Ford and Sidi, 1988; Sidi, 1991; Jbilou and Sadok, 1991; Sidi and Shapira, 1998; Sidi, 2008; Sidi, 2017a; Sidi, 2017b; Brezinski *et al.*, 2018; Brezinski *et al.*, 2020). Unfortunately, these methods do not extend well to nonlinear functions, especially due to conditioning problems (Sidi, 1986; Sidi and Bridger, 1988; Scieur *et al.*, 2016). Recent works have nevertheless proven the convergence of such methods, provided that good conditioning of the linear system (Sidi, 2019) can be ensured.

There are also other classes of nonlinear acceleration algorithms, based on existing algorithms, for accelerating the convergence of scalar sequences (Brezinski, 1975). For instance, the topological epsilon vector algorithm (TEA) extends the idea of the scalar ε-algorithm of (Wynn, 1956) to vectors.

4

Nesterov Acceleration

This section presents a systematic interpretation of the acceleration of the gradient method stemming from Nesterov's original work (Nesterov, 1983). The early parts of the section are devoted to the gradient method and the "optimized gradient method," due to Drori and Teboulle (2014) and Kim and Fessler (2016). The motivations and ideas underlying the latter are intuitive and very similar to those behind the introduction of Chebyshev methods for optimizing quadratic functions (see Section 2). Furthermore, the optimized gradient method has a relatively simple format and proof and can be used as an inspiration for developing numerous variants with wider ranges of applications, including Nesterov's early accelerated gradient methods (Nesterov, 1983; Nesterov, 2013) and the fast iterative shrinkage-thresholding algorithm (Beck and Teboulle, 2009a, FISTA). Although some parts of this section are more technical, we believe all the ideas can be reasonably well understood even when skipping, or skimming through the algebraic proofs. The section and the proofs are organized so that each time an additional ingredient (strong convexity, constraints, etc.) is included, its inclusion only requires a few additional ingredients compared to the previous (simpler) proofs of the base versions of the method.

We start with the theory and interpretation of acceleration in a simple setting: smooth unconstrained convex minimization in a Euclidean space. All subsequent developments follow from the same template, namely a linear combination of regularity inequalities, with additional ingredients being added one by one. The next part is devoted to methods that take advantage of strong convexity by using the same ideas and algorithmic structures. On the way, we provide a few different (equivalent) templates for the algorithms, since in more advanced settings, those templates do not generalize in the same way. We then recap and discuss a few practical extensions for handling constrained problems, nonsmooth regularization terms, unknown problem parameters/line-searches, and non-

Euclidean geometries. Finally, we briefly discuss a popular ordinary differential equation (ODE)-based interpretation of Nesterov's method. Techniques for obtaining the worst-case analyses presented throughout this text are presented in Appendix C, and notebooks for simpler reproduction of the proofs are provided in Section 4.9.

4.1 Introduction

In the first part of this section, we consider smooth unconstrained convex minimization problems. This type of problems is a direct extension of unconstrained convex quadratic minimization problems where the quadratic function has eigenvalues bounded above by some constant. More precisely, we consider the simple unconstrained differentiable convex minimization problem

$$f_\star = \min_{x \in \mathbb{R}^d} f(x), \tag{4.1}$$

where f is convex with an L-Lipschitz gradient (we call such functions convex and L-smooth, see Definition 4.1 below), and we assume throughout that there exists a minimizer $x_\star$. The goal of the methods presented below is to find a candidate solution x satisfying $f(x) - f_\star \leq \epsilon$ for some $\epsilon > 0$. Depending on the target application, other quality measures, such as guarantees on $\|\nabla f(x)\|_2$ or $\|x - x_\star\|_2$, might be preferred. We refer to Section 4.9 "Changing the performance measure", for discussions on this topic.

We start with the analysis of gradient descent and then show that its iteration complexity can be significantly improved using an acceleration technique proposed by Nesterov (1983).

After presenting the theory for the smooth convex case we see how it goes in the smooth strongly convex one. This class of problems extends to that of unconstrained convex quadratic minimization problems where the quadratic function has eigenvalues respectively bounded above and below by some constants L and μ.

Definition 4.1. Let $0 \leq \mu < L < +\infty$. A continuously differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is L-smooth and μ-strongly convex (denoted $f \in \mathcal{F}_{\mu,L}$) if and only if

- (L-smoothness) for all $x, y \in \mathbb{R}^d$, it holds that

$$f(x) \leq f(y) + \langle \nabla f(y); x - y \rangle + \frac{L}{2}\|x - y\|_2^2, \tag{4.2}$$

- (μ-strong convexity) for all $x, y \in \mathbb{R}^d$, it holds that

$$f(x) \geq f(y) + \langle \nabla f(y); x - y \rangle + \frac{\mu}{2}\|x - y\|_2^2. \tag{4.3}$$

Furthermore, we denote by $q = \frac{\mu}{L}$ the inverse condition number (that is, $q = \frac{1}{\kappa}$, with κ is the usual condition number as used e.g., in Section 2) of functions in the class $\mathcal{F}_{\mu,L}$.

Notation. We use the notation $\mathcal{F}_{0,L}$ for the set of smooth convex functions. By extension, we use $\mathcal{F}_{0,\infty}$ for the set of (possibly non-differentiable) proper closed convex functions (i.e., convex functions whose epigraphs are non-empty closed convex sets). Finally, we denote by $\partial f(x)$ the subdifferential of f at $x \in \mathbb{R}^d$ and by $g_f(x) \in \partial f(x)$ a particular subgradient of f at x.

Smooth strongly convex functions. Figure 4.1 provides an illustration of the global quadratic *upper approximation* (with curvature L) on $f(\cdot)$ due to smoothness and of the global quadratic *lower approximation* (with curvature μ) on $f(\cdot)$ due to strong convexity.

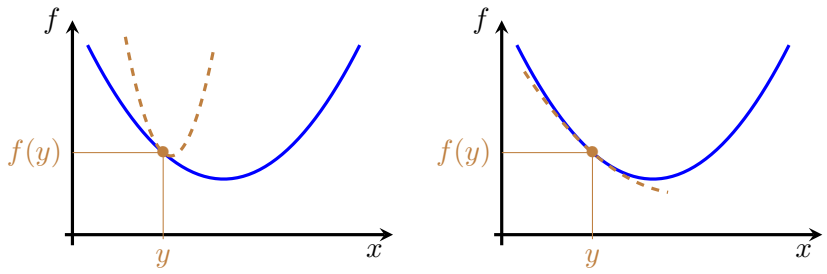

Figure 4.1: Let $f(\cdot)$ (blue) be a differentiable function. (Left) Smoothness: $f(\cdot)$ (blue) is L-smooth if and only if it is upper bounded by $f(y) + \langle \nabla f(y); . - y \rangle + \frac{L}{2}\|. - y\|_2^2$ (dashed, brown) for all y. (Right) Strong convexity: $f(\cdot)$ (blue) is μ-strongly convex if and only if it is lower bounded by $f(y) + \langle \nabla f(y); . - y \rangle + \frac{\mu}{2}\|. - y\|_2^2$ (dashed, brown) for all y.

A number of inequalities can be written to characterize functions in $\mathcal{F}_{\mu,L}$: see, for example, Nesterov (2003, Theorem 2.1.5). When analyzing methods for minimizing functions in this class, it is crucial to have the *right* inequalities at our disposal, as worst-case analyses essentially boil down to appropriately combining such inequalities. We provide the most important inequalities along with their interpretations and proofs in Appendix A. In this section, we only use three. First, we use the quadratic upper and lower bounds arising from the definition of smooth strongly convex functions, that is, (4.2) and (4.3). For some analyses however, we need an additional inequality, provided by the following theorem. This inequality is often referred to as an *interpolation* (or *extension*) inequality. Its proof is relatively simple: it only consists of requiring all quadratic lower bounds from (4.3) to be below all quadratic upper bounds from (4.2) (details in Appendix A.1). It can be shown that worst-case analyses of all first-order methods for minimizing smooth strongly convex functions can be performed using *only* this inequality for some specific values of x and y (details in Appendix C).

Theorem 4.1. Let f be a continuously differentiable function. f is L-smooth and μ-strongly convex (possibly with $\mu = 0$) if and only if for all $x, y \in \mathbb{R}^d$, it holds that

$$\begin{aligned} f(x) \geq f(y) + \langle \nabla f(y); x - y \rangle + \frac{1}{2L} \|\nabla f(y) - \nabla f(x)\|_2^2 \\ + \frac{\mu L}{2(L-\mu)} \|x - y - \tfrac{1}{L}(\nabla f(x) - \nabla f(y))\|_2^2. \end{aligned} \tag{4.4}$$

As discussed later in Section 4.3.3, this inequality has some flaws. Therefore, we only use (4.2) and (4.3) whenever possible.

Before continuing to the next section, we mention that both smoothness and strong convexity are strong assumptions. More generic assumptions are discussed in Section 6 to obtain improved rates under weaker assumptions.

4.2 Gradient Method and Potential Functions

In this section, we analyze gradient descent using the concept of *potential functions*. The resulting proofs are technically simple, although they might not seem to provide any direct intuition about the method at hand. We use the same ideas to analyze a few improvements on gradient descent before providing interpretations underlying this mechanism.

4.2.1 Gradient Descent

The simplest and probably most natural method for minimizing differentiable functions is gradient descent. It is often attributed to Cauchy (1847) and consists of iterating

$$x_{k+1} = x_k - \gamma_k \nabla f(x_k),$$

where γ_k is some step size. There are many different techniques for picking γ_k, the simplest of which is to set $\gamma_k = 1/L$, assuming L is known—otherwise, line-search techniques are typically used; see Section 4.7. Our present objective is to bound the number of iterations required by gradient descent to obtain an approximate minimizer x_k of f that satisfies $f(x_k) - f_\star \leq \epsilon$.

4.2.2 A Simple Proof Mechanism: Potential Functions

Potential (or Lyapunov/energy) functions are classical tools for proving convergence rates in the first-order literature, and a nice recent review of this topic is given by Bansal and Gupta (2019). For gradient descent, the idea consists in recursively using a simple inequality (proof below),

$$(k+1)(f(x_{k+1}) - f_\star) + \frac{L}{2}\|x_{k+1} - x_\star\|_2^2 \leq k(f(x_k) - f_\star) + \frac{L}{2}\|x_k - x_\star\|_2^2,$$

that is valid for all $f \in \mathcal{F}_{0,L}$ and all $x_k \in \mathbb{R}^d$ when $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$. In this context, we refer to

$$\phi_k \triangleq k(f(x_k) - f_\star) + \frac{L}{2}\|x_k - x_\star\|_2^2$$

as a potential and use $\phi_{k+1} \leq \phi_k$ as the building block for the worst-case analysis. Once such a *potential inequality* $\phi_{k+1} \leq \phi_k$ is established, a worst-case guarantee can easily be deduced through a recursive argument, yielding

$$N(f(x_N) - f_\star) \leq \phi_N \leq \phi_{N-1} \leq \ldots \leq \phi_0 = \frac{L}{2}\|x_0 - x_\star\|_2^2, \tag{4.5}$$

and hence, $f(x_N) - f_\star \leq \frac{L}{2N}\|x_0 - x_\star\|_2^2$. We also conclude that the worst-case accuracy of gradient descent is $O(N^{-1})$ or equivalently, that its iteration complexity is $O(\epsilon^{-1})$. Therefore, the main inequality to be proved for this worst-case analysis to work is the *potential* inequality $\phi_{k+1} \leq \phi_k$. In other words, the analysis of N iterations of gradient descent is reduced to the analysis of a single iteration, using an appropriate potential. This kind of approach was already used for example by Nesterov (1983), and many different variants of the potential function can be used to prove convergence of gradient descent and related methods in similar ways.

Theorem 4.2. Let f be an L-smooth convex function, $x_\star \in \operatorname{argmin}_x f(x)$, and $k \in \mathbb{N}$. For any $A_k \geq 0$ and $x_k \in \mathbb{R}^d$, it holds that

$$\begin{aligned} A_{k+1}(f(x_{k+1}) - f_\star) + \frac{L}{2}\|x_{k+1} - x_\star\|_2^2 \\ \leq A_k(f(x_k) - f_\star) + \frac{L}{2}\|x_k - x_\star\|_2^2, \end{aligned}$$

with $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$ and $A_{k+1} = 1 + A_k$.

Proof. The proof consists of performing a weighted sum of the following inequalities:

- convexity of f between x_k and $x_\star$, with weight $\lambda_1 = A_{k+1} - A_k$:

$$0 \geq f(x_k) - f_\star + \langle \nabla f(x_k); x_\star - x_k \rangle,$$

- smoothness of f between x_k and x_{k+1} with weight $\lambda_2 = A_{k+1}$:

$$0 \geq f(x_{k+1}) - \left(f(x_k) + \langle \nabla f(x_k); x_{k+1} - x_k \rangle + \frac{L}{2}\|x_k - x_{k+1}\|_2^2 \right).$$

  The last inequality is often referred to as the *descent lemma* since substituting x_{k+1} allows to obtain $f(x_{k+1}) \leq f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|_2^2$.

The weighted sum forms a valid inequality:

$$\begin{aligned} 0 \geq & \lambda_1[f(x_k) - f_\star + \langle \nabla f(x_k); x_\star - x_k \rangle] \\ & + \lambda_2[f(x_{k+1}) - (f(x_k) + \langle \nabla f(x_k); x_{k+1} - x_k \rangle + \frac{L}{2}\|x_k - x_{k+1}\|_2^2)]. \end{aligned}$$

Using $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$, this inequality can be rewritten (by completing the squares or simply extending both expressions and verifying that they match on a term-by-term basis) as follows:

$$\begin{aligned}0 \geq &(A_k+1)(f(x_{k+1}) - f_\star) + \frac{L}{2}\|x_{k+1} - x_\star\|_2^2 \\ &- A_k(f(x_k) - f_\star) - \frac{L}{2}\|x_k - x_\star\|_2^2 + \frac{A_{k+1}-1}{2L}\|\nabla f(x_k)\|_2^2 \\ &- (A_{k+1} - A_k - 1)\langle \nabla f(x_k); x_k - x_\star\rangle,\end{aligned}$$

which can be reorganized and simplified to

$$\begin{aligned}&(A_k+1)(f(x_{k+1}) - f_\star) + \frac{L}{2}\|x_{k+1} - x_\star\|_2^2 \\ \leq &A_k(f(x_k) - f_\star) + \frac{L}{2}\|x_k - x_\star\|_2^2 - \frac{A_{k+1}-1}{2L}\|\nabla f(x_k)\|_2^2 \\ &+ (A_{k+1} - A_k - 1)\langle \nabla f(x_k); x_k - x_\star\rangle \\ \leq &A_k(f(x_k) - f_\star) + \frac{L}{2}\|x_k - x_\star\|_2^2,\end{aligned}$$

where the last inequality follows from picking $A_{k+1} = A_k + 1$ and neglecting the last residual term $-\frac{A_k}{2L}\|\nabla f(x_k)\|_2^2$ (which is nonpositive) on the right-hand side. ■

A convergence rate for gradient descent can be obtained directly as a consequence of Theorem 4.2, following the reasoning of (4.5), and the worst-case guarantee corresponds to $f(x_N) - f_\star = O(A_N^{-1}) = O(N^{-1})$. We detail this in the next corollary.

Corollary 4.3. Let f be an L-smooth convex function, and $x_\star \in \operatorname{argmin}_x f(x)$. For any $N \in \mathbb{N}$, the iterates of gradient descent with step size $\gamma_0 = \gamma_1 = \ldots = \gamma_N = \frac{1}{L}$ satisfy

$$f(x_N) - f_\star \leq \frac{L\|x_0 - x_\star\|_2^2}{2N}.$$

Proof. Following the reasoning of (4.5), we recursively use Theorem 4.2, starting with $A_0 = 0$. That is, we define

$$\phi_k \triangleq A_k(f(x_k) - f_\star) + \frac{L}{2}\|x_k - x_\star\|_2^2$$

and recursively use the inequality $\phi_{k+1} \leq \phi_k$ from Theorem 4.2, with $A_{k+1} = A_k + 1$ and $A_0 = 0$; hence, $A_k = k$. We thus obtain

$$A_N(f(x_N) - f_\star) \leq \phi_N \leq \ldots \leq \phi_0 = \frac{L}{2}\|x_0 - x_\star\|_2^2,$$

resulting in the desired statement

$$f(x_N) - f_\star \leq \frac{L}{2A_N}\|x_0 - x_\star\|_2^2 = \frac{L}{2N}\|x_0 - x_\star\|_2^2. \quad ■$$

4.2.3 How Conservative is this Worst-case Guarantee?

Before moving to other methods, we show that the worst-case rate $O(N^{-1})$ of gradient descent is attained on very simple problems, motivating the search for alternate methods with better guarantees. This rate is observed on, e.g., all functions that are nearly linear over large regions. One such common function is the Huber loss (with $x_\star = 0$, arbitrarily):

$$f(x) = \begin{cases} a_\tau |x| - b_\tau & \text{if } |x| \geq \tau, \\ \frac{L}{2} x^2 & \text{otherwise,} \end{cases}$$

with $a_\tau = L\tau$ and $b_\tau = -\frac{L}{2}\tau^2$ to ensure its continuity and differentiability. On this function, as long as the iterates of gradient descent satisfy $|x_k| \geq \tau$, they behave as if the function were linear, and the gradient is constant. It is therefore relatively easy to explicitly compute all iterates. In particular, by picking $\tau = \frac{|x_0|}{2N+1}$, we get $f(x_N) - f_\star = \frac{L\|x_0 - x_\star\|_2^2}{2(2N+1)}$ and reach the $O(N^{-1})$ worst-case bound; see Drori and Teboulle (2014, Theorem 3.2). Therefore, it appears that the worst-case bound from Corollary 4.3 for gradient descent can only be improved in terms of the constants, but the rate itself is the best possible one for this simple method; see, for example, (Drori and Teboulle, 2014; Drori, 2014) for the corresponding tight expressions.

In the next section, we show that similar reasoning based on potential functions produces methods with improved worst-case convergence rate $O(N^{-2})$, compared to the $O(N^{-1})$ of vanilla gradient descent.

4.3 Optimized Gradient Method

Given that the complexity bound for gradient descent cannot be improved, it is reasonable to look for alternate, hopefully better, methods. In this section, we show that accelerated methods can be designed by optimizing their worst-case performance. To do so, we start with a reasonably broad family of candidate first-order methods described by

$$\begin{aligned} y_1 &= y_0 - h_{1,0}\nabla f(y_0), \\ y_2 &= y_1 - h_{2,0}\nabla f(y_0) - h_{2,1}\nabla f(y_1), \\ y_3 &= y_2 - h_{3,0}\nabla f(y_0) - h_{3,1}\nabla f(y_1) - h_{3,2}\nabla f(y_2), \\ &\vdots \\ y_N &= y_{N-1} - \sum_{i=0}^{N-1} h_{N,i}\nabla f(y_i). \end{aligned} \tag{4.6}$$

Of course, methods in this form are impractical since they require keeping track of all previous gradients. Neglecting this potential problem for now, one possibility for choosing the step size $\{h_{i,j}\}$ is to solve a minimax problem:

$$\min_{\{h_{i,j}\}} \max_{f \in \mathcal{F}_{0,L}} \left\{ \frac{f(y_N) - f_\star}{\|y_0 - x_\star\|_2^2} \,:\, y_N \text{ obtained from (4.6) and } y_0 \right\}. \tag{4.7}$$

In other words, we are looking for the best possible worst-case ratio among methods of the form (4.6). Of course, different target notions of accuracy could be considered instead of $(f(y_N) - f_\star)/\|y_0 - x_\star\|_2^2$, but we proceed with this notion for now.

It turns out that (4.7) has a clean solution, obtained by Kim and Fessler (2016), based on clever reformulations and relaxations of (4.7) developed by Drori and Teboulle (2014) (some details are provided in Section 4.9). Furthermore, this method has "factorized" forms that do not require keeping track of previous gradients. The optimized gradient method (OGM) is parameterized by a sequence $\{\theta_{k,N}\}_k$ that is constructed recursively starting from $\theta_{-1,N} = 0$ (or equivalently $\theta_{0,N} = 1$), using

$$\theta_{k+1,N} = \begin{cases} \frac{1+\sqrt{4\theta_{k,N}^2+1}}{2} & \text{if } k \leq N-2 \\ \frac{1+\sqrt{8\theta_{k,N}^2+1}}{2} & \text{if } k = N-1. \end{cases} \tag{4.8}$$

We also mention that optimized gradient methods can be stated in various equivalent formats, we provide two variants in Algorithm 9 and Algorithm 10 (a rigorous equivalence statement is provided in Appendix B.1.1). While the shape of Algorithm 10 is more common in accelerated methods, the equivalent formulation provided in Algorithm 9 allows for slightly more direct proofs.

Algorithm 9 Optimized gradient method (OGM), form I

Input: L-smooth convex function f, initial point x_0, and budget N.

1: **Initialize** $z_0 = y_0 = x_0$ and $\theta_{-1,N} = 0$.
2: **for** $k = 0, \ldots, N-1$ **do**
3: $\quad \theta_{k,N} = \frac{1+\sqrt{4\theta_{k-1,N}^2+1}}{2}$
4: $\quad y_k = \left(1 - \frac{1}{\theta_{k,N}}\right) x_k + \frac{1}{\theta_{k,N}} z_k$
5: $\quad x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k)$
6: $\quad z_{k+1} = x_0 - \frac{2}{L}\sum_{i=0}^{k} \theta_{i,N} \nabla f(y_k)$
7: **end for**

Output: Approximate solution $y_N = \left(1 - \frac{1}{\theta_{N,N}}\right) x_N + \frac{1}{\theta_{N,N}} z_N$ with $\theta_{N,N} = \frac{1+\sqrt{8\theta_{N-1,N}^2+1}}{2}$.

Direct approaches to (4.7) are rather technical—see details in (Drori and Teboulle, 2014; Kim and Fessler, 2016). However, showing that the OGM is indeed optimal on the class of smooth convex functions can be accomplished indirectly by providing an upper bound on its worst-case complexity guarantees and by showing that no first-order method can have a better worst-case guarantee on this class of problems. We detail a fully explicit worst-case guarantee for OGM in the next section. It consists in showing that

$$\phi_k \triangleq 2\theta_{k-1,N}^2 \left(f(y_{k-1}) - f_\star - \frac{1}{2L}\|\nabla f(y_{k-1})\|_2^2\right) + \frac{L}{2}\|z_k - x_\star\|_2^2 \tag{4.9}$$

is a potential function for the optimized gradient method when $k < N$ (Theorem 4.4, below). For $k = N$, we need a minor adjustment (Lemma 4.5,

Algorithm 10 Optimized gradient method (OGM), form II

Input: L-smooth convex function f, initial point x_0, and budget N.

1: **Initialize** $z_0 = y_0 = x_0$ and $\theta_{0,N} = 1$.
2: **for** $k = 0, \dots, N-1$ **do**
3: $\quad \theta_{k+1,N} = \begin{cases} \frac{1+\sqrt{4\theta_{k,N}^2+1}}{2} & \text{if } k \leq N-2 \\ \frac{1+\sqrt{8\theta_{k,N}^2+1}}{2} & \text{if } k = N-1. \end{cases}$
4: $\quad x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k)$
5: $\quad y_{k+1} = x_{k+1} + \frac{\theta_{k,N}-1}{\theta_{k+1,N}}(x_{k+1} - x_k) + \frac{\theta_{k,N}}{\theta_{k+1,N}}(x_{k+1} - y_k)$.
6: **end for**

Output: Approximate solution y_N.

below) to obtain a bound on $f(y_N) - f_\star$ and not in terms of $f(y_N) - f_\star - \frac{1}{2L}\|\nabla f(y_N)\|_2^2$, which appears in the potential.

As in the case of gradient descent, the proof relies on potential functions. Following the recursive argument from (4.5), the convergence guarantee is driven by the convergence speed of $\theta_{k,N}^{-2}$ towards 0. We note that when $k < N-1$,

$$\theta_{k+1,N} = \frac{1+\sqrt{4\theta_{k,N}^2+1}}{2} \geq \frac{1+2\theta_{k,N}}{2} = \theta_{k,N} + \frac{1}{2}, \tag{4.10}$$

and therefore, $\theta_{k,N} \geq \frac{k}{2} + 1$. We also directly obtain

$$\theta_{N,N} = \frac{1+\sqrt{8\theta_{N-1,N}^2+1}}{2} \geq \frac{1+\sqrt{2(N+1)^2+1}}{2} \geq \frac{N+1}{\sqrt{2}}, \tag{4.11}$$

and hence, $\theta_{N,N}^{-2} = O(N^{-2})$. Before providing the proof, we mention that it heavily relies on inequality (4.4) with $\mu = 0$. This inequality is key for formulating (4.7) in a tractable way.

4.3.1 A Potential for the Optimized Gradient Method

The main point now is to prove that (4.9) is indeed a potential for the optimized gradient method. We emphasize again that our main motivation for proving this is to show that the OGM provides a good template algorithm for acceleration (i.e., a method involving two or three sequences) and that the corresponding potential functions can also be used as a template for the analysis of more advanced methods.

Note that the potential structure does not seem immediately intuitive: it was actually found using computer-assisted proof design techniques; see Section 4.9 "On obtaining the proofs in this section" and Appendix C for further references. In particular, the following theorem can be found in Taylor and Bach (2019, Theorem 11).

Theorem 4.4. Let f be an L-smooth convex function, $x_\star \in \operatorname{argmin}_x f(x)$, and $N \in \mathbb{N}$. For any $k \in \mathbb{N}$ with $0 \leq k \leq N-1$ and any $y_{k-1}, z_k \in \mathbb{R}^d$, it holds that

$$\begin{aligned} 2\theta_{k,N}^2 \left(f(y_k) - f_\star - \frac{1}{2L}\|\nabla f(y_k)\|_2^2 \right) + \frac{L}{2}\|z_{k+1} - x_\star\|_2^2 \\ \leq 2\theta_{k-1,N}^2 \left(f(y_{k-1}) - f_\star - \frac{1}{2L}\|\nabla f(y_{k-1})\|_2^2 \right) + \frac{L}{2}\|z_k - x_\star\|_2^2, \end{aligned}$$

when y_k and z_{k+1} are obtained from Algorithm 9.

Proof. Recall that the algorithm can be written as

$$\begin{aligned} y_k &= \left(1 - \frac{1}{\theta_{k,N}}\right)\left(y_{k-1} - \frac{1}{L}\nabla f(y_{k-1})\right) + \frac{1}{\theta_{k,N}} z_k \\ z_{k+1} &= z_k - \frac{2\theta_{k,N}}{L}\nabla f(y_k). \end{aligned}$$

The proof consists of performing a weighted sum of the following inequalities.

- Smoothness and convexity of f between y_{k-1} and y_k with weight $\lambda_1 = 2\theta_{k-1,N}^2$:

$$\begin{aligned} 0 \geq & f(y_k) - f(y_{k-1}) + \langle \nabla f(y_k); y_{k-1} - y_k \rangle \\ & + \frac{1}{2L}\|\nabla f(y_k) - \nabla f(y_{k-1})\|_2^2. \end{aligned}$$

- Smoothness and convexity of f between $x_\star$ and y_k with weight $\lambda_2 = 2\theta_{k,N}$:

$$0 \geq f(y_k) - f_\star + \langle \nabla f(y_k); x_\star - y_k \rangle + \frac{1}{2L}\|\nabla f(y_k)\|_2^2.$$

Since the weights are nonnegative, the weighted sum produces a valid inequality:

$$\begin{aligned} 0 \geq & \lambda_1 \Big[f(y_k) - f(y_{k-1}) + \langle \nabla f(y_k); y_{k-1} - y_k \rangle + \frac{1}{2L}\|\nabla f(y_k) \\ & - \nabla f(y_{k-1})\|_2^2 \Big] \\ & + \lambda_2 \left[f(y_k) - f_\star + \langle \nabla f(y_k); x_\star - y_k \rangle + \frac{1}{2L}\|\nabla f(y_k)\|_2^2 \right], \end{aligned} \tag{4.12}$$

which (either by completing the squares or simply by extending both expressions and verifying that they match on a term-by-term basis) can be reformulated as

$$\begin{aligned} 0 \geq & \lambda_1 \Big[f(y_k) - f(y_{k-1}) + \langle \nabla f(y_k); y_{k-1} - y_k \rangle \\ & + \frac{1}{2L}\|\nabla f(y_k) - \nabla f(y_{k-1})\|_2^2 \Big] \\ & + \lambda_2 \left[f(y_k) - f_\star + \langle \nabla f(y_k); x_\star - y_k \rangle + \frac{1}{2L}\|\nabla f(y_k)\|_2^2 \right] \\ = & 2\theta_{k,N}^2 \left(f(y_k) - f_\star - \frac{1}{2L}\|\nabla f(y_k)\|_2^2 \right) + \frac{L}{2}\|z_{k+1} - x_\star\|_2^2 \\ & - 2\theta_{k-1,N}^2 \left(f(y_{k-1}) - f_\star - \frac{1}{2L}\|\nabla f(y_{k-1})\|_2^2 \right) - \frac{L}{2}\|z_k - x_\star\|_2^2 \end{aligned}$$

$$+\frac{2}{\theta_{k,N}}\left(\theta_{k-1,N}^2+\theta_{k,N}-\theta_{k,N}^2\right)\langle\nabla f(y_k);y_{k-1}-\tfrac{1}{L}\nabla f(y_{k-1})-z_k\rangle$$
$$+2\left(\theta_{k-1,N}^2+\theta_{k,N}-\theta_{k,N}^2\right)\left(f(y_k)-f_\star+\frac{1}{2L}\|\nabla f(y_k)\|_2^2\right).$$

The desired conclusion follows from picking $\theta_{k,N}\geq\theta_{k-1,N}$ satisfying

$$\theta_{k-1,N}^2+\theta_{k,N}-\theta_{k,N}^2=0,$$

and hence the choice (4.8), thus reaching

$$2\theta_{k,N}^2\left(f(y_k)-f_\star-\frac{1}{2L}\|\nabla f(y_k)\|_2^2\right)+\frac{L}{2}\|z_{k+1}-x_\star\|_2^2$$
$$\leq 2\theta_{k-1,N}^2\left(f(y_{k-1})-f_\star-\frac{1}{2L}\|\nabla f(y_{k-1})\|_2^2\right)+\frac{L}{2}\|z_k-x_\star\|_2^2.\ \blacksquare$$

A final technical fix is required now. To show that the optimized gradient method is an optimal solution to (4.7), we need an upper bound on the function values, rather than on the function values minus a squared gradient norm. This discrepancy is handled by the following technical lemma.

Lemma 4.5. Let f be an L-smooth convex function, $x_\star\in\operatorname{argmin}_x f(x)$, and $N\in\mathbb{N}$. For any $y_{N-1},z_N\in\mathbb{R}^d$, it holds that

$$\theta_{N,N}^2\left(f(y_N)-f_\star\right)+\frac{L}{2}\|z_N-\tfrac{\theta_{N,N}}{L}\nabla f(y_N)-x_\star\|_2^2$$
$$\leq 2\theta_{N-1,N}^2\left(f(y_{N-1})-f_\star-\frac{1}{2L}\|\nabla f(y_{N-1})\|_2^2\right)+\frac{L}{2}\|z_N-x_\star\|_2^2,$$

where y_N is obtained from Algorithm 9.

Proof. The proof consists of performing a weighted sum of the following inequalities.

- Smoothness and convexity of f between y_{N-1} and y_N with weight $\lambda_1=2\theta_{N-1,N}^2$:

$$0\geq f(y_N)-f(y_{N-1})+\langle\nabla f(y_N);y_{N-1}-y_N\rangle$$
$$+\frac{1}{2L}\|\nabla f(y_N)-\nabla f(y_{N-1})\|_2^2.$$

- Smoothness and convexity of f between $x_\star$ and y_N with weight $\lambda_2=\theta_{N,N}$:

$$0\geq f(y_N)-f_\star+\langle\nabla f(y_N);x_\star-y_N\rangle+\frac{1}{2L}\|\nabla f(y_N)\|_2^2.$$

Since the weights are nonnegative, the weighted sum produces a valid inequality:

$$0\geq\lambda_1\left[f(y_N)-f(y_{N-1})+\langle\nabla f(y_N);y_{N-1}-y_N\rangle\right.$$
$$\left.+\frac{1}{2L}\|\nabla f(y_N)-\nabla f(y_{N-1})\|_2^2\right]$$
$$+\lambda_2\left[f(y_N)-f_\star+\langle\nabla f(y_N);x_\star-y_N\rangle+\frac{1}{2L}\|\nabla f(y_N)\|_2^2\right],$$

which can be reformulated as

$$\begin{aligned}0 \geq & \theta_{N,N}^2 \left(f(y_N) - f_\star\right) + \frac{L}{2}\|z_N - \tfrac{\theta_{N,N}}{L}\nabla f(y_N) - x_\star\|_2^2 \\ & - 2\theta_{N-1,N}^2 \left(f(y_{N-1}) - f_\star - \frac{1}{2L}\|\nabla f(y_{N-1})\|_2^2\right) - \frac{L}{2}\|z_N - x_\star\|_2^2 \\ & + \frac{1}{\theta_{N,N}}\left(2\theta_{N-1,N}^2 - \theta_{N,N}^2 + \theta_{N,N}\right) \\ & \quad \times \langle \nabla f(y_N); y_{N-1} - \tfrac{1}{L}\nabla f(y_{N-1}) - z_N\rangle \\ & + \left(2\theta_{N-1,N}^2 - \theta_{N,N}^2 + \theta_{N,N}\right)\left(f(y_N) - f_\star + \frac{1}{2L}\|\nabla f(y_N)\|_2^2\right).\end{aligned}$$

The conclusion follows from choosing $\theta_{N,N} \geq \theta_{N-1,N}$ such that

$$2\theta_{N-1,N}^2 - \theta_{N,N}^2 + \theta_{N,N} = 0,$$

thereby reaching the desired inequality:

$$\begin{aligned}& \theta_{N,N}^2 \left(f(y_N) - f_\star\right) + \frac{L}{2}\|z_N - \tfrac{\theta_{N,N}}{L}\nabla f(y_N) - x_\star\|_2^2 \\ & \quad \leq 2\theta_{N-1,N}^2 \left(f(y_{N-1}) - f_\star - \frac{1}{2L}\|\nabla f(y_{N-1})\|_2^2\right) + \frac{L}{2}\|z_N - x_\star\|_2^2. \quad \blacksquare\end{aligned}$$

By combining Theorem 4.4 and the technical Lemma 4.5, we get the final worst-case performance bound of the OGM on function values, detailed in the corollary below.

Corollary 4.6. Let f be an L-smooth convex function, and $x_\star \in \operatorname{argmin}_x f(x)$. For any $N \in \mathbb{N}$ and $x_0 \in \mathbb{R}^d$, the output of the optimized gradient method (OGM, Algorithm 9 or Algorithm 10) satisfies

$$f(y_N) - f_\star \leq \frac{L\|x_0 - x_\star\|_2^2}{2\theta_{N,N}^2} \leq \frac{L\|x_0 - x_\star\|_2^2}{(N+1)^2}.$$

Proof. Defining, for $k \in \{1, \ldots, N\}$

$$\phi_k \triangleq 2\theta_{k-1,N}^2 \left(f(y_{k-1}) - f_\star - \frac{1}{2L}\|\nabla f(y_{k-1})\|_2^2\right) + \frac{L}{2}\|z_k - x_\star\|_2^2,$$

and

$$\phi_{N+1} \triangleq \theta_{N,N}^2 \left(f(y_N) - f_\star\right) + \frac{L}{2}\|z_N - \tfrac{\theta_{N,N}}{L}\nabla f(y_N) - x_\star\|_2^2,$$

we reach the desired statement:

$$\theta_{N,N}^2 (f(y_N) - f_\star) \leq \phi_{N+1} \leq \phi_N \leq \ldots \leq \phi_0 = \frac{L}{2}\|x_0 - x_\star\|_2^2,$$

using Theorem 4.4 and technical Lemma 4.5. We obtain the last bound by using $\theta_{N,N} \geq (N+1)/\sqrt{2}$; see (4.11). $\blacksquare$

In the following section, we mostly use potential functions, relying directly on the function value $f(x_k)$ instead of $f(y_k)$ for practical reasons discussed below. Note that using the *descent lemma* (i.e., the inequality $f(x_{k+1}) \leq f(y_k) - \frac{1}{2L}\|\nabla f(y_k)\|_2^2$) directly on the potential function allows us to obtain a bound on $f(x_{k+1})$ for the OGM. This result can be found in (Kim and Fessler, 2017, Theorem 3.1) without the "potential function" mechanism.

4.3.2 Optimality of Optimized Gradient Methods

A nice, commonly used guide for designing optimal methods consists of constructing problems that are difficult for all methods within a certain class. This strategy results in *lower complexity bounds*, and it is often deployed via the concept of *minimax risk* (of a class of problems and a class of methods)—see, e.g., Guzmán and Nemirovsky (2015)—which corresponds to the worst-case performance of the best method within the prescribed class. In this section, we briefly discuss such results in the context of smooth convex minimization, on the particular class of *black-box* first-order methods. The term *black-box* is used to emphasize that the method has no prior knowledge of f (beyond the class of functions to which f belongs, so methods are allowed to use L) and that it can only obtain information about f by a evaluating its gradient/function value through an *oracle*.

Of particular interest to us, Drori (2017) established that the worst-case performance achieved by the optimized gradient method (see Corollary 4.6) on the class of smooth convex functions cannot in general be improved by any black-box first-order method.

Theorem 4.7. (Drori, 2017, Theorem 3) Let $L > 0$, $d, N \in \mathbb{N}$ with $d \geq N + 1$. For any black-box first-order method that performs at most N calls to the first-order oracle $(f(\cdot), \nabla f(\cdot))$, there exists a function $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$ and $x_0 \in \mathbb{R}^d$ such that

$$f(x_N) - f(x_\star) \geq \frac{L\|x_0 - x_\star\|_2^2}{2\theta_{N,N}^2},$$

with $x_\star \in \operatorname{argmin}_x f(x)$, x_N is the output of the method under consideration, and x_0 its input.

In the previous sections, we showed that $\theta_{N,N}^2 \geq \frac{(N+1)^2}{2}$. It is also relatively easy to establish that

$$\theta_{k+1,N} \leq \frac{1 + \sqrt{(2\theta_{k,N} + 1)^2}}{2} = 1 + \theta_{k,N},$$

thereby obtaining $\theta_{k,N} \leq k + 1$ (because $\theta_{0,N} = 1$) as well as

$$\theta_{N,N} = \frac{1 + \sqrt{8\theta_{N-1,N}^2 + 1}}{2} \leq \frac{1 + \sqrt{8N^2 + 1}}{2} \leq \sqrt{2}N + 1.$$

We conclude (through Theorem 4.7) that the lower bound has the form

$$f(x_N) - f(x_\star) \geq \frac{L\|x_0 - x_\star\|_2^2}{2\theta_{N,N}^2} \geq \frac{L\|x_0 - x_\star\|_2^2}{(\sqrt{2}N + 1)^2} = \Omega(N^{-2}).$$

While Drori's approach to obtaining this lower bound is rather technical (a slightly simplified and weaker version of this result can be found in (Drori and Taylor, 2021, Corollary 5)), there are simpler approaches that allow us to show that the rate $\Omega(N^{-2})$ (that is, neglecting the tight constants) cannot in general be beaten in black-box smooth convex minimization. For one such example, we refer to (Nesterov, 2003, Theorem 2.1.6).

In a closely related line of work, (Nemirovsky, 1991) established similar *exact* bounds in the context of solving linear systems of equations and for minimizing convex quadratic functions (see also Section 2.3.4). For convex quadratic problems whose Hessian has bounded eigenvalues between 0 and L, these lower bounds are attained by the Chebyshev (see Section 2) and by conjugate gradient methods (Nemirovsky, 1991; Nemirovsky, 1992).

Perhaps surprisingly, the conjugate gradient method also achieves the lower complexity bound of smooth convex minimization provided by Theorem 4.7. Furthermore, the proof follows essentially the same structure as that for the OGM. In particular, it relies on the same potential function (see Appendix B.2).

4.3.3 Optimized Gradient Method: Summary

Before going further, we quickly summarize what we have learned from the optimized gradient method. First of all, the optimized gradient method can be seen as a counterpart of the Chebyshev method for minimizing quadratics, applied to smooth convex minimization. It is an *optimal* method in the sense that it has the smallest possible worst-case ratio $\frac{f(y_N)-f_\star}{\|y_0-x_\star\|_2^2}$ over the class $f \in \mathcal{F}_{0,L}$ among all black-box first-order methods, given a fixed computational budget of N gradient evaluations. Furthermore, although this method has a few drawbacks (we mention a few below), it can be seen as a *template* for designing other accelerated methods using the same algorithmic and proof structures. We extensively use variants of this template below. In other words, most variants of *accelerated gradient methods* rely on the same two (or three) sequence structures, and on similar potential functions. Such variants usually rely on slight variations in the choice of the parameters used throughout the iterative process, typically involving less aggressive step size strategies (i.e., smaller values for $\{h_{i,j}\}$ in (4.6)).

Second, the OGM is not a very practical method as such: it is fined-tuned for unconstrained smooth convex minimization and does not readily extend to other situations, such as situations involving constraints, for which (4.4) does not hold in general; see the discussions in (Drori, 2018) and Remark A.1.

On the other hand, we see in what follows that it is relatively easy to design other methods that follow the same template and achieve the same $O(N^{-2})$ rate, while resolving the issues of the OGM listed above. Such methods use slightly less aggressive step size strategies, at the cost of being slightly suboptimal for (4.7), i.e., they have slightly worse worst-case guarantees. In this vein, we start by discussing the original accelerated gradient method, proposed by Nesterov (1983).

4.4 Nesterov's Acceleration

Motivated by the format of the optimized gradient method, we detail a potential-based proof for Nesterov's method. We then quickly review the concept of *estimate sequences* and show that they provide an interpretation of potential functions as increasingly good models of the function to be minimized. Finally, we extend these results to strongly

convex minimization.

4.4.1 Nesterov's Method, from Potential Functions

In this section, we follow the algorithmic template provided by the optimized gradient method. In this spirit, we start by discussing the first accelerated method in its simplest form (Algorithm 11) as well as its potential function, originally proposed by Nesterov (1983), but the presentation here is different.

Our goal is to derive the simplest algebraic proof for this scheme. We follow the algorithmic template of the optimized gradient method (which is further motivated in Section 4.6.1). Once a potential is chosen, the proofs are quite straightforward as simple combinations of inequalities and basic algebra. Our choice of potential function is not immediately obvious but allows for simple extensions afterwards. Other choices are possible, for example, incorporating $f(y_k)$ as (in the OGM) or additional terms such as $\|\nabla f(x_k)\|_2^2$. We pick a potential function similar to that used for gradient descent and that of the optimized gradient method, which is written

$$\phi_k \triangleq A_k(f(x_k) - f_\star) + \frac{L}{2}\|z_k - x_\star\|_2^2,$$

where one iteration of the algorithm has the following form, reminiscent of the OGM:

$$\begin{aligned} y_k &= x_k + \tau_k(z_k - x_k) \\ x_{k+1} &= y_k - \alpha_k \nabla f(y_k) \\ z_{k+1} &= z_k - \gamma_k \nabla f(y_k). \end{aligned} \tag{4.13}$$

Our goal is to select algorithmic parameters $\{(\tau_k, \alpha_k, \gamma_k)\}_k$ so as to greedily make A_{k+1} as large as possible as a function of A_k since the convergence rate of the method is controlled by the inverse of the growth rate of A_k, i.e., $f(x_N) - f_\star = O(A_N^{-1})$.

In practice, we can pick $A_{k+1} = A_k + \frac{1}{2}(1 + \sqrt{4A_k + 1})$ by choosing $\tau_k = 1 - A_k/A_{k+1}$, $\alpha_k = \frac{1}{L}$, and $\gamma_k = (A_{k+1} - A_k)/L$ (see Algorithm 11), and the proof is then quite compact.

Algorithm 11 Nesterov's method, form I

Input: An L-smooth convex function f and initial point x_0.

1: **Initialize** $z_0 = x_0$ and $A_0 = 0$.
2: **for** $k = 0, \ldots$ **do**
3: $\quad a_k = \frac{1}{2}(1 + \sqrt{4A_k + 1})$
4: $\quad A_{k+1} = A_k + a_k$
5: $\quad y_k = x_k + (1 - \frac{A_k}{A_{k+1}})(z_k - x_k)$
6: $\quad x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k)$
7: $\quad z_{k+1} = z_k - \frac{A_{k+1} - A_k}{L}\nabla f(y_k)$
8: **end for**

Output: Approximate solution x_{k+1}.

Before continuing to the proof of the potential inequality, we show that $A_k^{-1} = O(k^{-2})$. Indeed, we have,

$$A_k = A_{k-1} + \frac{1+\sqrt{4A_{k-1}+1}}{2} \geq A_{k-1} + \frac{1}{2} + \sqrt{A_{k-1}} \geq \left(\sqrt{A_{k-1}} + \frac{1}{2}\right)^2 \geq \frac{k^2}{4}, \tag{4.14}$$

where the last inequality follows from a recursive application of the previous one, along with $A_0 = 0$.

Theorem 4.8. Let f be an L-smooth convex function, $x_\star \in \operatorname{argmin}_x f(x)$, and $k \in \mathbb{N}$. For any $x_k, z_k \in \mathbb{R}^d$ and $A_k \geq 0$, the iterates of Algorithm 11 satisfy

$$A_{k+1}(f(x_{k+1}) - f_\star) + \frac{L}{2}\|z_{k+1} - x_\star\|_2^2 \leq A_k(f(x_k) - f_\star) + \frac{L}{2}\|z_k - x_\star\|_2^2,$$

with $A_{k+1} = A_k + \frac{1+\sqrt{4A_k+1}}{2}$.

Proof. The proof consists of a weighted sum of the following inequalities.

- Convexity of f between $x_\star$ and y_k with weight $\lambda_1 = A_{k+1} - A_k$:
$$f_\star \geq f(y_k) + \langle \nabla f(y_k); x_\star - y_k \rangle.$$
- Convexity of f between x_k and y_k with weight $\lambda_2 = A_k$:
$$f(x_k) \geq f(y_k) + \langle \nabla f(y_k); x_k - y_k \rangle.$$
- Smoothness of f between y_k and x_{k+1} (a.k.a., *descent lemma*) with weight $\lambda_3 = A_{k+1}$:
$$f(y_k) + \langle \nabla f(y_k); x_{k+1} - y_k \rangle + \frac{L}{2}\|x_{k+1} - y_k\|_2^2 \geq f(x_{k+1}).$$

We therefore arrive at the following valid inequality

$$\begin{aligned} 0 \geq & \lambda_1[f(y_k) - f_\star + \langle \nabla f(y_k); x_\star - y_k \rangle] \\ & + \lambda_2[f(y_k) - f(x_k) + \langle \nabla f(y_k); x_k - y_k \rangle] \\ & + \lambda_3[f(x_{k+1}) - f(y_k) - \langle \nabla f(y_k); x_{k+1} - y_k \rangle - \frac{L}{2}\|x_{k+1} - y_k\|_2^2]. \end{aligned}$$

For the sake of simplicity, we do not substitute A_{k+1} by its expression until the last stage of the reformulation. Substituting y_k, x_{k+1}, and z_{k+1} by their expressions in (4.13) along with $\tau_k = 1 - A_k/A_{k+1}$, $\alpha_k = \frac{1}{L}$, and $\gamma_k = \frac{A_{k+1}-A_k}{L}$, basic algebra shows that the previous inequality can be reorganized as

$$\begin{aligned} 0 \geq & A_{k+1}(f(x_{k+1}) - f_\star) + \frac{L}{2}\|z_{k+1} - x_\star\|_2^2 \\ & - A_k(f(x_k) - f_\star) - \frac{L}{2}\|z_k - x_\star\|_2^2 \\ & + \frac{A_{k+1} - (A_k - A_{k+1})^2}{2L}\|\nabla f(y_k)\|_2^2. \end{aligned}$$

The claim follows from selecting $A_{k+1} \geq A_k$ such that $A_{k+1} - (A_k - A_{k+1})^2 = 0$, thereby reaching

$$A_{k+1}(f(x_{k+1}) - f_\star) + \frac{L}{2}\|z_{k+1} - x_\star\|_2^2 \leq A_k(f(x_k) - f_\star) + \frac{L}{2}\|z_k - x_\star\|_2^2. \quad \blacksquare$$

The final worst-case guarantee is obtained by using the same chaining argument as in (4.5), combined with an upper bound on A_N.

Corollary 4.9. Let f be an L-smooth convex function, and $x_\star \in \operatorname{argmin}_x f(x)$. For any $N \in \mathbb{N}$, the iterates of Algorithm 11 satisfy

$$f(x_N) - f_\star \leq \frac{2L\|x_0 - x_\star\|_2^2}{N^2}.$$

Proof. Following the argument of (4.5), we recursively use Theorem 4.8 with $A_0 = 0$:

$$A_N(f(x_N) - f_\star) \leq \phi_N \leq \ldots \leq \phi_0 = \frac{L}{2}\|x_0 - x_\star\|_2^2,$$

which yields

$$f(x_N) - f_\star \leq \frac{L\|x_0 - x_\star\|_2^2}{2A_N} \leq \frac{2L\|x_0 - x_\star\|_2^2}{N^2},$$

where we used $A_N \geq N^2/4$ from (4.14) to reach the last inequality. $\blacksquare$

Before moving on, we emphasize that the rate of $O(N^{-2})$ matches that of lower bounds (see, e.g., Theorem 4.7) up to absolute constants.

Finally, note that Nesterov's method is often written in a slightly different format, similar to that of Algorithm 10. The alternate formulation omits the third sequence z_k and is provided in Algorithm 12. It is preferred in many references on the topic due to its simplicity. A third equivalent variant is provided in Algorithm 13; this variant turns out to be useful when generalizing the method beyond Euclidean spaces. The equivalence statements between Algorithm 11, Algorithm 12, and Algorithm 13 are relatively simple and are provided in Appendix B.1.2. Many references tend to favor one of these formulations, and we want to point out that they are equivalent in the base problem setup of unconstrained smooth convex minimization. Although the expression of the different formats in terms of the same external sequence $\{A_k\}_k$ does not always correspond to their simplest forms (i.e., alternate parameterizations might be simpler, particularly in the strongly convex case which follows), we proceed with this sequence to avoid introducing too many variations on the same theme.

4.4.2 Estimate Sequence Interpretation

We now relate the potential function approach to *estimate sequences*. That is, we relate acceleration to first-order methods maintaining a model of the function throughout the iterative procedure. This approach was originally developed in (Nesterov, 2003, Section 2.2), and it has since been used in numerous works to obtain accelerated first-order methods in various settings (see discussions in Section 4.9). We present a slightly

Algorithm 12 Nesterov's method, form II

Input: An L-smooth convex function f and initial point x_0.

Initialize $y_0 = x_0$ and $A_0 = 0$.
for $k = 0, \ldots$ **do**
  $a_k = \frac{1}{2}(1 + \sqrt{4A_k + 1})$
  $A_{k+1} = A_k + a_k$
  $x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k)$
  $y_{k+1} = x_{k+1} + \frac{a_k - 1}{a_{k+1}}(x_{k+1} - x_k)$
end for

Output: Approximate solution x_{k+1}.

Algorithm 13 Nesterov's method, form III

Input: An L-smooth convex function f and initial point x_0.

Initialize $z_0 = x_0$ and $A_0 = 0$.
for $k = 0, \ldots$ **do**
  $a_k = \frac{1}{2}(1 + \sqrt{4A_k + 1})$
  $A_{k+1} = A_k + a_k$
  $y_k = x_k + (1 - \frac{A_k}{A_{k+1}})(z_k - x_k)$
  $z_{k+1} = z_k - \frac{A_{k+1} - A_k}{L}\nabla f(y_k)$
  $x_{k+1} = \frac{A_k}{A_{k+1}}x_k + (1 - \frac{A_k}{A_{k+1}})z_{k+1}$
end for

Output: Approximate solution x_{k+1}.

modified version, related to those of (Baes, 2009; Wilson *et al.*, 2021), which simplifies our comparisons with the previous material.

Estimate Sequences

As we see below, the basic idea underlying estimate sequences is closely related to that of potential functions, but it has explicit interpretations in terms of models of the objective function f. More precisely a sequence of pairs $\{(A_k, \varphi_k(x))\}_k$, with $A_k \geq 0$ and $\varphi_k : \mathbb{R}^d \to \mathbb{R}$, is called an estimate sequence of a function f if
(i) for all $k \geq 0$ and $x \in \mathbb{R}^d$ we have

$$\varphi_k(x) - f(x) \leq A_k^{-1}(\varphi_0(x) - f(x)), \tag{4.15}$$

(ii) $A_k \to \infty$ as $k \to \infty$.
If in addition, an estimate sequence satisfies
(iii) for all $k \geq 0$, there exists some x_k such that $f(x_k) \leq \varphi_k(x_\star)$, then we can guarantee that $f(x_k) - f_\star = O(A_k^{-1})$.

The purpose of estimate sequences is to start from an initial model $\varphi_0(x)$ satisfying $\varphi_0(x) \geq f_\star$ for all $x \in \mathbb{R}^d$ and then to design a sequence of convex models φ_k that

are increasingly good approximations of f, in the sense of (4.15). We provide further comments on conditions (i) and (iii), assuming for simplicity that $\{A_k\}$ is monotonically increasing (as is the case for all methods treated in this section).

- Regarding (i), for all $x \in \mathbb{R}^d$, we have to design φ_k to be either (a) a lower bound on the function (i.e., $\varphi_k(x) - f(x) \leq 0$ for that x) or (b) an increasingly good upper approximation of $f(x)$ when $0 \leq \varphi_k(x) - f(x) \leq A_k^{-1}(\varphi_0(x) - f(x))$ for that x. That is, we require that the error $|f(x) - \varphi_k(x)|$, incurred when approximating $f(x)$ by $\varphi_k(x)$, gets smaller for all x for which $\varphi_k(x)$ is an upper bound on $f(x)$.

  To develop such models and the corresponding methods, three sequences of points are commonly used: (a) minimizers of our models φ_k that correspond to iterates z_k of the corresponding method; (b) a sequence y_k of points, whose first-order information is used to update the model of the function; and (c) the iterates x_k, corresponding to the best possible $f(x_k)$ that we can form. (The iteratives often do not correspond to the minimum of the model, φ_k, which is not necessarily an upper bound on the function.)

- Regarding (iii), this condition ensures that the models φ_k remain upper bounds on the optimal value $f_\star$. That is, it ensures that $f_\star \leq \varphi_k(x_\star)$ (since $f_\star \leq f(x_k)$) and hence that $\varphi_k(x_\star) - f_\star \geq 0$. From previous bullet point, this ensures that the modeling error of $f_\star$ goes to 0 asymptotically as k increases. More formally, conditions (ii) and (iii) allow us to construct proofs similar to potential functions and to obtain convergence rates. That is, under (iii), we get that

$$f(x_k) - f_\star \leq \varphi_k(x_\star) - f(x_\star) \leq A_k^{-1}(\varphi_0(x_\star) - f(x_\star)), \tag{4.16}$$

  and that therefore $f(x_k) - f_\star \leq O(A_k^{-1})$. The convergence rate is thereby dictated by the rate of A_k^{-1}, which goes to 0 by (ii).

Now, the game consists of picking appropriate sequences $\{(A_k, \varphi_k)\}$ that correspond to simple algorithms. We thus translate our potential function results in terms of estimate sequences.

Potential Functions as Estimate Sequences

One can observe that potential functions and estimate sequences are closely related. First, in both cases, the convergence speed is dictated by that of a scalar sequence A_k^{-1}. In fact, there is one subtle but important difference between the two approaches: whereas $\varphi_k(x)$ should be an increasingly good approximation of f for all x in the context of estimate sequences, potential functions require a model to be an increasingly good approximation of only $f_\star$, which is less restrictive. Hence, estimate sequences are more general but may not effectively handle situations in which the analysis actually requires having a weaker model that holds only on $f_\star$, and not of $f(x)$, for all x. We make this discussion more concrete via three examples, namely gradient descent, Nesterov's method, and the optimized gradient method.

- Gradient descent: the potential inequality from Theorem 4.2 actually holds for all x, and not only $x_\star$, as the proof does not exploit the optimality of $x_\star$. That is, it is proved that:

$$(A_k+1)(f(x_{k+1})-f(x))+\frac{L}{2}\|x_{k+1}-x\|_2^2 \\ \leq A_k(f(x_k)-f(x))+\frac{L}{2}\|x_k-x\|_2^2$$

  for all $x\in\mathbb{R}^d$. Therefore, the pair $\{(A_k,\varphi_k(x))\}_k$ with

$$\varphi_k(x)=f(x_k)+\frac{L}{2A_k}\|x_k-x\|_2^2$$

  and $A_k=A_0+k$ (with $A_0>0$) is an estimate sequence for gradient descent.

- Nesterov's first method: the potential inequality from Theorem 4.8 also holds for all $x\in\mathbb{R}^d$, not only $x_\star$, as the proof does not exploit the optimality of $x_\star$. That is, it is proved that:

$$A_{k+1}(f(x_{k+1})-f(x))+\frac{L}{2}\|z_{k+1}-x\|_2^2 \\ \leq A_k(f(x_k)-f(x))+\frac{L}{2}\|z_k-x\|_2^2$$

  for all $x\in\mathbb{R}^d$. Hence, the pair $\{(A_k,\varphi_k(x))\}_k$ with

$$\varphi_k(x)=f(x_k)+\frac{L}{2A_k}\|z_k-x\|_2^2$$

  and $A_k=A_{k-1}+\frac{1+\sqrt{4A_{k-1}+1}}{2}$ (with $A_0>0$) is an estimate sequence for Nesterov's method.

- Optimized gradient method: the potential inequality from Theorem 4.4 exploits the fact that $x_\star$ is an optimal point. Indeed, the proof relies on

$$f(x_\star)\geq f(y_k)+\langle\nabla f(y_k);x_\star-y_k\rangle+\frac{1}{2L}\|\nabla f(y_k)\|_2^2,$$

  which is an instance of Equation (4.4) exploiting $\nabla f(x_\star)=0$. This does not mean that there is no *estimate sequence*-type model of the function as the algorithm proceeds, but the potential does not directly correspond to one. Alternatively, one can interpret

$$\varphi_k(x)=f(y_k)-\frac{1}{2L}\|\nabla f(y_k)\|_2^2+\frac{L}{4\theta_{k,N}^2}\|z_{k+1}-x\|_2^2$$

  as an increasingly good model of $f_\star$ (i.e., it is an increasingly good approximation of $f(x)$ for all x such that $\nabla f(x)=0$).

  A similar conclusion holds for the conjugate gradient method (CG), from Appendix B.2. We are not aware of any estimate sequence that can be used to prove that CG reaches the lower bound from Theorem 4.7.

These discussions can be extended to the strongly convex setting, which we now address.

4.5 Acceleration under Strong Convexity

Before designing faster methods that exploit strong convexity, we briefly describe the benefits and limitations of this additional assumption. Roughly speaking, strong convexity guarantees that the gradient gets larger further away from the optimal solution. One way of looking at it is as follows: a function f is L-smooth and μ-strongly convex if and only if there exists some $(L-\mu)$-smooth convex function $\tilde{f}$ such that

$$f(x) = \tilde{f}(x) + \frac{\mu}{2}\|x - x_\star\|_2^2,$$

where $x_\star$ is an optimal point for both f and $\tilde{f}$. Therefore, one iteration of gradient descent can be described as follows:

$$\begin{aligned} x_{k+1} - x_\star &= x_k - x_\star - \gamma \nabla f(x_k) \\ &= x_k - x_\star - \gamma(\nabla \tilde{f}(x_k) + \mu(x_k - x_\star)) \\ &= (1 - \gamma\mu)(x_k - x_\star) - \gamma \nabla \tilde{f}(x_k). \end{aligned}$$

We see that for sufficiently small step sizes γ, there is an additional *contraction* effect due to the factor $(1-\gamma\mu)$, as compared to the effect that gradient descent has on smooth convex functions such as $\tilde{f}$. In what follows, we adapt our proofs to develop accelerated methods in the strongly convex case. Because the smooth strongly convex functions are sandwiched between two quadratic functions, these assumptions are of course much more restrictive than smoothness alone.

4.5.1 Gradient Descent and Strong Convexity

As in the smooth convex case, the smooth strongly convex case can be studied through potential functions. There are many ways to prove convergence rates for this setting, but we only consider one that allows us to recover the $\mu = 0$ case as its limit such that the results are well-defined even in degenerate cases. The next proof is essentially the same as that for the smooth convex case in Theorem 4.2, and the same inequalities are used, with strong convexity instead of convexity. The potential is only slightly modified, thereby allowing A_k to have a geometric growth rate:

$$\phi_k \triangleq A_k(f(x_k) - f_\star) + \frac{L + \mu A_k}{2}\|x_k - x_\star\|_2^2.$$

For notational convenience, we use $q = \frac{\mu}{L}$ to denote the inverse condition ratio. This quantity plays a key role in the geometric convergence of first-order methods in the presence of strong convexity.

Theorem 4.10. Let f be an L-smooth μ-strongly (possibly with $\mu = 0$) convex function, $x_\star \in \operatorname{argmin}_x f(x)$, and $k \in \mathbb{N}$. For any $A_k \geq 0$ and any x_k, it holds that

$$\begin{aligned} A_{k+1}(f(x_{k+1}) - f_\star) + \frac{L + \mu A_{k+1}}{2}\|x_{k+1} - x_\star\|_2^2 \\ \leq A_k(f(x_k) - f_\star) + \frac{L + \mu A_k}{2}\|x_k - x_\star\|_2^2, \end{aligned}$$

with $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$, $A_{k+1} = (1 + A_k)/(1 - q)$, and $q = \frac{\mu}{L}$.

Proof. The proof consists of performing a weighted sum of the following inequalities.

- Strong convexity of f between x_k and $x_\star$, with weight $\lambda_1 = A_{k+1} - A_k$:
$$0 \geq f(x_k) - f_\star + \langle \nabla f(x_k); x_\star - x_k \rangle + \frac{\mu}{2}\|x_\star - x_k\|_2^2.$$
- Smoothness of f between x_k and x_{k+1} with weight $\lambda_2 = A_{k+1}$
$$0 \geq f(x_{k+1}) - f(x_k) - \langle \nabla f(x_k); x_{k+1} - x_k \rangle - \frac{L}{2}\|x_k - x_{k+1}\|_2^2.$$

This weighted sum yields a valid inequality:

$$\begin{aligned} 0 \geq & \lambda_1[f(x_k) - f_\star + \langle \nabla f(x_k); x_\star - x_k \rangle + \frac{\mu}{2}\|x_\star - x_k\|_2^2] \\ & + \lambda_2[f(x_{k+1}) - f(x_k) - \langle \nabla f(x_k); x_{k+1} - x_k \rangle - \frac{L}{2}\|x_k - x_{k+1}\|_2^2]. \end{aligned}$$

Using $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$, this inequality can be rewritten exactly as

$$\begin{aligned} & A_{k+1}(f(x_{k+1}) - f_\star) + \frac{L + \mu A_{k+1}}{2}\|x_{k+1} - x_\star\|_2^2 \\ \leq & A_k(f(x_k) - f_\star) + \frac{L + \mu A_k}{2}\|x_k - x_\star\|_2^2 \\ & - \frac{(1-q)A_{k+1} - 1}{2L}\|\nabla f(x_k)\|_2^2 \\ & + ((1-q)A_{k+1} - A_k - 1)\langle \nabla f(x_k); x_k - x_\star \rangle. \end{aligned}$$

The desired inequality follows from $A_{k+1} = (1 + A_k)/(1 - q)$ and the sign of A_k, making one of the last two terms nonpositive and the other equal to zero, thus reaching

$$\begin{aligned} A_{k+1}(f(x_{k+1}) - f_\star) + \frac{L + \mu A_{k+1}}{2}\|x_{k+1} - x_\star\|_2^2 & \\ \leq A_k(f(x_k) - f_\star) + \frac{L + \mu A_k}{2}\|x_k - x_\star\|_2^2. & \blacksquare \end{aligned}$$

From this theorem, we observe that adding strong convexity to the problem allows A_k to follow a geometric rate given by $(1 - q)^{-1}$ (where we again denote by $q = \frac{\mu}{L}$ the inverse condition number). The corresponding iteration complexity of gradient descent to find an approximate solution $f(x_k) - f_\star \leq \epsilon$ for smooth strongly convex minimization is therefore $O(\frac{L}{\mu}\log\frac{1}{\epsilon})$. This rate is essentially tight, as can be verified on quadratic functions (see, e.g., Section 2), and it follows from the following corollary whose result can be translated to iteration complexity using the same arguments as in Corollary 2.2.

Corollary 4.11. Let f be an L-smooth μ-strongly convex function, and $x_\star \in \operatorname{argmin}_x f(x)$. For any $N \in \mathbb{N}$, the iterates of gradient descent with step size $\gamma_0 = \gamma_1 = \ldots = \gamma_N = \frac{1}{L}$ satisfy

$$f(x_N) - f_\star \leq \frac{\mu\|x_0 - x_\star\|_2^2}{2((1-q)^{-N} - 1)},$$

with the inverse condition number $q = \frac{\mu}{L}$.

Proof. Following the reasoning of (4.5), we recursively use Theorem 4.2 starting with $A_0 = 0$; that is,

$$A_N(f(x_N) - f_\star) \leq \phi_N \leq \ldots \leq \phi_0 = \frac{L}{2}\|x_0 - x_\star\|_2^2,$$

and we notice that the recurrence equation $A_{k+1} = (A_k + 1)/(1 - q)$ has the solution $A_k = ((1 - q)^{-k} - 1)/q$. The final bound is obtained by using $f(x_N) - f_\star \leq \frac{L\|x_0 - x_\star\|_2^2}{2A_N}$ again. ■

Note that as $\mu \to 0$, the result of Corollary 4.11 tends to that of Corollary 4.3.

Remark 4.1 (Lower bounds)**.** As in the smooth convex case, one can derive lower complexity bounds for smooth strongly convex optimization. Using the lower bounds from smooth strongly convex quadratic minimization (for which Chebyshev's methods have optimal iteration complexity), one can conclude that no black-box first-order method can behave better than $f(x_k) - f_\star = O(\rho^k)$ with $\rho = \frac{(1-\sqrt{q})^2}{(1+\sqrt{q})^2}$ (see Section 2). In other words, lower complexity bounds from the quadratic optimization setting have the form $f(x_k) - f_\star = \Omega(\rho^k)$. We refer the reader to Nesterov (2003) and Nemirovsky (1992) for more details.

For smooth strongly convex problems beyond quadratics, this lower bound can be improved to $f(x_k) - f_\star = \Omega((1 - \sqrt{q})^{2k})$ as provided in (Drori and Taylor, 2021, Corollary 4). In this context, we see that Nesterov's acceleration satisfies

$$f(x_k) - f_\star = O((1 - \sqrt{q})^k).$$

That is, it has an $O(\sqrt{\frac{L}{\mu}} \log \frac{1}{\epsilon})$ iteration complexity (using similar simplifications as those of Corollary 2.2), reaching the lower complexity bound up to a constant factor. As for the optimized gradient method provided in Section 4.3, an optimal method for the smooth strongly convex case is detailed in Section 4.6.1, and it can be shown to match exactly the corresponding worst-case lower complexity bound.

4.5.2 Acceleration for Smooth Strongly Convex Objectives

To adapt our proofs of convergence of accelerated methods to the strongly convex case, we need to make a small adjustment to the shape of the previous accelerated method

$$\begin{aligned} y_k &= x_k + \tau_k(z_k - x_k) \\ x_{k+1} &= y_k - \alpha_k \nabla f(y_k) \\ z_{k+1} &= (1 - \tfrac{\mu}{L}\delta_k)z_k + \tfrac{\mu}{L}\delta_k y_k - \gamma_k \nabla f(y_k). \end{aligned} \tag{4.17}$$

As discussed below, there is an optimized gradient method for smooth strongly convex minimization, similar to OGM for the smooth convex setting (see Section 4.3), with this structure (details in Section 4.6.1). Following this scheme, Nesterov's method for strongly convex problems is presented in Algorithm 14. As in the smooth convex case,

Algorithm 14 Nesterov's method, form I

Input: An L-smooth μ-strongly (possibly with $\mu = 0$) convex function f and initial x_0.

Initialize $z_0 = x_0$ and $A_0 = 0$; $q = \mu/L$ (inverse condition ratio).
for $k = 0, \dots$ **do**
  $A_{k+1} = \frac{2A_k+1+\sqrt{4A_k+4qA_k^2+1}}{2(1-q)}$ $\{A_{k+1}$ solution to $(A_k - A_{k+1})^2 - A_{k+1} - qA_{k+1}^2 = 0\}$
  Set $\tau_k = \frac{(A_{k+1}-A_k)(1+qA_k)}{A_{k+1}+2qA_kA_{k+1}-qA_k^2}$ and $\delta_k = \frac{A_{k+1}-A_k}{1+qA_{k+1}}$
  $y_k = x_k + \tau_k(z_k - x_k)$
  $x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k)$
  $z_{k+1} = (1 - q\delta_k)z_k + q\delta_k y_k - \frac{\delta_k}{L}\nabla f(y_k)$
end for

Output: Approximate solution x_{k+1}.

we detail several of its convenient reformulations in Algorithm 28 and Algorithm 29. The corresponding equivalences are established in Appendix B.1.3.

Regarding the potential, we make the same adjustment as for gradient descent, arriving to the following theorem.

Theorem 4.12. Let f be an L-smooth μ-strongly (possibly with $\mu = 0$) convex function, $x_\star \in \operatorname{argmin}_x f(x)$, and $k \in \mathbb{N}$. For all $x_k, z_k \in \mathbb{R}^d$ and $A_k \geq 0$, the iterates of Algorithm 14 satisfy

$$\begin{aligned} A_{k+1}(f(x_{k+1}) - f_\star) + \frac{L + \mu A_{k+1}}{2}\|z_{k+1} - x_\star\|_2^2 \\ \leq A_k(f(x_k) - f_\star) + \frac{L + \mu A_k}{2}\|z_k - x_\star\|_2^2, \end{aligned}$$

with $A_{k+1} = \frac{2A_k+1+\sqrt{4A_k+4qA_k^2+1}}{2(1-q)}$ and $q = \frac{\mu}{L}$.

Proof. The proof consists of a weighted sum of the following inequalities.

- Strong convexity between $x_\star$ and y_k with weight $\lambda_1 = A_{k+1} - A_k$:

$$f_\star \geq f(y_k) + \langle \nabla f(y_k); x_\star - y_k \rangle + \frac{\mu}{2}\|x_\star - y_k\|_2^2.$$

- Convexity between x_k and y_k with weight $\lambda_2 = A_k$:

$$f(x_k) \geq f(y_k) + \langle \nabla f(y_k); x_k - y_k \rangle.$$

- Smoothness between y_k and x_{k+1} (*descent lemma*) with weight $\lambda_3 = A_{k+1}$

$$f(y_k) + \langle \nabla f(y_k); x_{k+1} - y_k \rangle + \frac{L}{2}\|x_{k+1} - y_k\|_2^2 \geq f(x_{k+1}).$$

We therefore arrive at the following valid inequality:

$$\begin{aligned}0 \geq &\lambda_1[f(y_k) - f_\star + \langle \nabla f(y_k); x_\star - y_k\rangle + \frac{\mu}{2}\|x_\star - y_k\|_2^2] \\ &+ \lambda_2[f(y_k) - f(x_k) + \langle \nabla f(y_k); x_k - y_k\rangle] \\ &+ \lambda_3[f(x_{k+1}) - f(y_k) - \langle \nabla f(y_k); x_{k+1} - y_k\rangle - \frac{L}{2}\|x_{k+1} - y_k\|_2^2].\end{aligned}$$

For the sake of simplicity, we do not substitute A_{k+1} by its expression until the last stage of the reformulation. After substituting x_{k+1}, z_{k+1} by their expressions in (4.17) along with $\tau_k = \frac{(A_{k+1}-A_k)(1+qA_k)}{A_{k+1}+2qA_kA_{k+1}-qA_k^2}$, $\alpha_k = \frac{1}{L}$, $\delta_k = \frac{A_{k+1}-A_k}{1+qA_{k+1}}$, and $\gamma_k = \frac{\delta_k}{L}$, basic algebra shows that the previous inequality can be reorganized as

$$\begin{aligned}&A_{k+1}(f(x_{k+1}) - f_\star) + \frac{L+\mu A_{k+1}}{2}\|z_{k+1} - x_\star\|_2^2 \\ \leq &A_k(f(x_k) - f_\star) + \frac{L+\mu A_k}{2}\|z_k - x_\star\|_2^2 \\ &+ \frac{(A_k - A_{k+1})^2 - A_{k+1} - qA_{k+1}^2}{1+qA_{k+1}}\frac{1}{2L}\|\nabla f(y_k)\|_2^2 \\ &- A_k^2\frac{(A_{k+1}-A_k)(1+qA_k)(1+qA_{k+1})}{(A_{k+1}+2qA_kA_{k+1}-qA_k^2)^2}\frac{\mu}{2}\|x_k - z_k\|_2^2.\end{aligned}$$

The desired statement follows from selecting $A_{k+1} \geq A_k \geq 0$ such that

$$(A_k - A_{k+1})^2 - A_{k+1} - qA_{k+1}^2 = 0,$$

thus yielding

$$\begin{aligned}&A_{k+1}(f(x_{k+1}) - f_\star) + \frac{L+\mu A_{k+1}}{2}\|z_{k+1} - x_\star\|_2^2 \\ \leq &A_k(f(x_k) - f_\star) + \frac{L+\mu A_k}{2}\|z_k - x_\star\|_2^2. \quad \blacksquare\end{aligned}$$

The final worst-case guarantee is obtained by using the same reasoning as before, together with a simple bound on A_{k+1}:

$$\begin{aligned}A_{k+1} &= \frac{2A_k + 1 + \sqrt{4A_k + 4qA_k^2 + 1}}{2(1-q)} \\ &\geq \frac{2A_k + \sqrt{(2A_k\sqrt{q})^2}}{2(1-q)} = \frac{A_k}{1-\sqrt{q}},\end{aligned} \tag{4.18}$$

which means $f(x_k) - f_\star = O((1-\sqrt{q})^k)$ when $\mu > 0$, or alternatively that $O\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\epsilon}\right)$ is the iteration complexity of obtaining an approximate solution $f(x_N) - f_\star \leq \epsilon$ (using similar simplifications as those of Corollary 2.2). The following corollary summarizes our result for Nesterov's method.

Corollary 4.13. Let f be an L-smooth μ-strongly (possibly with $\mu = 0$) convex function and $x_\star \in \operatorname{argmin}_x f(x)$. For all $N \in \mathbb{N}$, $N \geq 1$, the iterates of Algorithm 14 satisfy

$$f(x_N) - f_\star \leq \min\left\{\frac{2}{N^2}, (1-\sqrt{q})^N\right\} L\|x_0 - x_\star\|_2^2,$$

with $q = \frac{\mu}{L}$.

Proof. Following the argument of (4.5), we recursively use Theorem 4.12 with $A_0 = 0$, together with the bounds on A_N for the smooth convex case (4.14) and for the smooth strongly convex one (4.18). (Note that A_{k+1} is an increasing function of μ, and hence the bound for the smooth case remains valid in the smooth strongly convex one.) We have $A_1 = \frac{1}{1-q} = \frac{1}{(1-\sqrt{q})(1+\sqrt{q})} \geq \frac{1}{2}(1-\sqrt{q})^{-1}$, thus reaching $A_N \geq \frac{1}{2}(1-\sqrt{q})^{-N}$. ∎

Remark 4.2. Before moving to the next section, we mention that another direct consequence of the potential inequality above (Theorem 4.12) is that z_k may also serve as an approximate solution to $x_\star$ when $\mu > 0$. Indeed, by using the inequality

$$\frac{L + \mu A_N}{2}\|z_N - x_\star\|_2^2 \leq \phi_N \leq \ldots \leq \phi_0 = \frac{L}{2}\|x_0 - x_\star\|_2^2,$$

it follows that

$$\|z_N - x_\star\|_2^2 \leq \frac{1}{1 + qA_N}\|x_0 - x_\star\|_2^2 \leq \frac{2(1-\sqrt{q})^N}{2(1-\sqrt{q})^N + q}\|x_0 - x_\star\|_2^2,$$

and hence that $\|z_N - x_\star\|_2^2 = O((1-\sqrt{q})^N)$. Therefore it also follows that

$$f(z_N) - f_\star \leq \frac{L}{2}\|z_N - x_\star\|_2^2 = O((1-\sqrt{q})^N).$$

In addition, since y_N is a convex combination of x_N and z_N, the same conclusion holds for $\|y_N - x_\star\|_2^2$ and $f(y_N) - f_\star$. Similar observations also apply to other variants of accelerated methods when $\mu > 0$.

4.5.3 A Simplified Stationary Method with Constant Momentum

Important simplifications are often made to the Nesterov's method in the strongly convex case where $\mu > 0$. Several approaches produce the same method, known as the "constant momentum" version of Nesterov's accelerated gradient. We derive this version by observing that the asymptotic (or stationary) behavior of Algorithm 14 can be characterized explicitly. In particular, when $k \to \infty$, it is clear that $A_k \to \infty$ as well. We can thus take the limits of all parameters as $A_k \to \infty$, to obtain a corresponding "limit/stationary method." This is similar in spirit to the result showing that Polyak's heavy-ball method is the asymptotic version of Chebyshev's method, discussed in Section 2.3.3. First, the convergence rate is obtained as

$$\lim_{A_k \to \infty} \frac{A_{k+1}}{A_k} = (1-\sqrt{q})^{-1}.$$

By taking the limits of all the algorithmic parameters, that is,

$$\lim_{A_k \to \infty} \tau_k = \frac{\sqrt{q}}{1+\sqrt{q}}, \quad \lim_{A_k \to \infty} \delta_k = \frac{1}{\sqrt{q}},$$

we obtain Algorithm 15 and its equivalent, probably most well-known, second form, provided as Algorithm 16.

Algorithm 15 Nesterov's method, form I, constant momentum

Input: L-smooth μ-strongly convex function f and initial point x_0.

Initialize $z_0 = x_0$ and $A_0 > 0$; $q = \mu/L$ (inverse condition ratio).
for $k = 0, \ldots$ **do**
  $A_{k+1} = \frac{A_k}{1-\sqrt{q}}$ {Only for the proof/relation to previous methods.}
  $y_k = x_k + \frac{\sqrt{q}}{1+\sqrt{q}}(z_k - x_k)$
  $x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k)$
  $z_{k+1} = (1-\sqrt{q})\, z_k + \sqrt{q}\left(y_k - \frac{1}{\mu}\nabla f(y_k)\right)$
end for

Output: Approximate solutions (y_k, x_{k+1}, z_{k+1}).

Algorithm 16 Nesterov's method, form II, constant momentum

Input: L-smooth μ-strongly convex function f and initial point x_0.

Initialize $y_0 = x_0$ and $A_0 > 0$; $q = \mu/L$ (inverse condition ratio).
for $k = 0, \ldots$ **do**
  $A_{k+1} = \frac{A_k}{1-\sqrt{q}}$ {Only for the proof/relation to previous methods.}
  $x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k)$
  $y_{k+1} = x_{k+1} + \frac{1-\sqrt{q}}{1+\sqrt{q}}(x_{k+1} - x_k)$
end for

Output: Approximate solutions (y_k, x_{k+1}).

From a worst-case analysis perspective, these simplifications correspond to using a Lyapunov function obtained by dividing the potential function of Theorem 4.12 by A_k and then taking the limit of the inequality:

$$\rho^{-1}\left(f(x_{k+1}) - f_\star + \frac{\mu}{2}\|z_{k+1} - x_\star\|_2^2\right) \leq f(x_k) - f_\star + \frac{\mu}{2}\|z_k - x_\star\|_2^2,$$

with $\rho = (1-\sqrt{q})$.

Theorem 4.14. Let f be an L-smooth μ-strongly convex function, $x_\star = \operatorname{argmin}_x f(x)$ and $k \in \mathbb{N}$. For any $x_k, z_k \in \mathbb{R}^d$, the iterates of Algorithm 15 (or equivalently those of Algorithm 16) satisfy

$$\rho^{-1}\left(f(x_{k+1}) - f_\star + \frac{\mu}{2}\|z_{k+1} - x_\star\|_2^2\right) \leq f(x_k) - f_\star + \frac{\mu}{2}\|z_k - x_\star\|_2^2.$$

Proof. Let $A_k > 0$ and $A_{k+1} = A_k/(1-\sqrt{q})$. The proof is essentially the same as that of as Theorem 4.12. That is, the weights used in this proof are those used in Theorem 4.12 divided by A_k, leading to a slight variation in the reformulation of the weighted sum, and the following valid inequality:

$$\begin{aligned}0 \geq &\lambda_1[f(y_k) - f_\star + \langle \nabla f(y_k); x_\star - y_k\rangle + \frac{\mu}{2}\|x_\star - y_k\|_2^2] \\ &+ \lambda_2[f(y_k) - f(x_k) + \langle \nabla f(y_k); x_k - y_k\rangle] \\ &+ \lambda_3[f(x_{k+1}) - f(y_k) - \langle \nabla f(y_k); x_{k+1} - y_k\rangle - \frac{L}{2}\|x_{k+1} - y_k\|_2^2],\end{aligned}$$

with weights

$$\begin{aligned}\lambda_1 &= \frac{A_{k+1} - A_k}{A_k} = \frac{\sqrt{q}}{1-\sqrt{q}}, \\ \lambda_2 &= \frac{A_k}{A_k} = 1, \text{ and} \\ \lambda_3 &= \frac{A_{k+1}}{A_k} = (1-\sqrt{q})^{-1}.\end{aligned}$$

By substituting

$$\begin{aligned}y_k &= x_k + \frac{\sqrt{q}}{1+\sqrt{q}}(z_k - x_k) \\ x_{k+1} &= y_k - \frac{1}{L}\nabla f(y_k) \\ z_{k+1} &= (1-\sqrt{q})\, z_k + \sqrt{q}\left(y_k - \frac{1}{\mu}\nabla f(y_k)\right),\end{aligned}$$

we arrive at the following valid inequality:

$$\begin{aligned}&\rho^{-1}\left(f(x_{k+1}) - f_\star + \frac{\mu}{2}\|z_{k+1} - x_\star\|_2^2\right) \\ &\quad\leq f(x_k) - f_\star + \frac{\mu}{2}\|z_k - x_\star\|_2^2 - \frac{\sqrt{q}}{(1+\sqrt{q})^2}\frac{\mu}{2}\|x_k - z_k\|_2^2.\end{aligned}$$

We reach the desired statement from the last term being nonpositive:

$$\rho^{-1}\left(f(x_{k+1}) - f_\star + \frac{\mu}{2}\|z_{k+1} - x_\star\|_2^2\right) \leq f(x_k) - f_\star + \frac{\mu}{2}\|z_k - x_\star\|_2^2. \quad \blacksquare$$

Corollary 4.15. Let f be an L-smooth μ-strongly convex function and $x_\star \in \operatorname{argmin}_x f(x)$. For all $N \in \mathbb{N}$, $N \geq 1$, the iterates of Algorithm 15 (or equivalently of Algorithm 16) satisfy

$$f(x_N) - f_\star \leq (1-\sqrt{q})^N\left(f(x_0) - f_\star + \frac{\mu}{2}\|x_0 - x_\star\|_2^2\right),$$

with $q = \frac{\mu}{L}$.

Proof. The desired result directly follows from Theorem 4.14 with

$$\phi_k \triangleq \rho^{-k}\left(f(x_k) - f_\star + \frac{\mu}{2}\|z_k - x_\star\|_2^2\right),$$

$\rho = 1-\sqrt{q}$, $z_0 = x_0$, and $\rho^{-N}(f(x_N) - f_\star) \leq \phi_N$. $\blacksquare$

Remark 4.3. In view of Section 4.4.2, one can also find estimate sequence interpretations of Algorithms 14 and 15 from their respective potential functions.

Remark 4.4. A few works on accelerated methods focus on understanding this particular instance of Nesterov's method. Our analysis here is largely inspired by that of Nesterov (2003), but such potentials can be obtained in different ways, see, for example (Wilson *et al.*, 2021; Shi *et al.*, 2021; Bansal and Gupta, 2019).

4.6 Recent Variants of Accelerated Methods

In this section, we first push the reasoning in terms of *potential functions* to its limit. We present the *information-theoretic exact method* (Taylor and Drori, 2021), which generalizes the optimized gradient descent in the strongly convex case. Similar to Nesterov's method with constant momentum, the *information-theoretic exact method* has a limit case that is known as the *triple momentum method* (Van Scoy *et al.*, 2017). We then discuss a more geometric variant, known as *geometric descent* (Bubeck *et al.*, 2015) or *quadratic averaging* (Drusvyatskiy *et al.*, 2018).

4.6.1 An Optimal Method for Strongly Convex Minimization

It turns out that there also exist optimal gradient methods for smooth strongly convex minimization that are similar to the optimized gradient method for smooth convex minimization. Such methods can be obtained by solving a minimax problem similar to (4.7) with different objectives.

The following scheme is optimal for the criterion $\frac{\|z_N - x_\star\|_2^2}{\|x_0 - x_\star\|_2^2}$, reaching the exact worst-case lower complexity bound for this criterion, as discussed below. In addition, this method reduces to the OGM (see Section 4.3) when $\mu = 0$ by using the correspondence $A_{k+1} = 4\theta_{k,N}^2$ (for $k < N$). Therefore, this method is *doubly optimal*, i.e., optimal according to two criteria, in the sense that it also achieves the lower complexity bound for $\frac{f(y_N) - f_\star}{\|x_0 - x_\star\|_2^2}$ when $\mu = 0$, using the last iteration adjustment from Lemma 4.5.

The following analysis is reminiscent of Nesterov's method in Algorithm 14 but also of the optimized gradient method and its proof (see Theorem 4.4). That is, the known potential function for the information-theoretic exact method (ITEM) relies on inequality (4.4), not only for its proof but also simply to simply show that it is nonnegative, which follows from instantiating (4.4) at $y = x_\star$. The following analyses can be found almost verbatim in (Taylor and Drori, 2021). The main proof of this section is particularly algebraic, but it can be reasonably skipped as it follows from similar ideas found in previous developments.

Theorem 4.16. Let f be an L-smooth μ-strongly (possibly with $\mu = 0$) convex function, $x_\star \in \operatorname{argmin}_x f(x)$ and $k \in \mathbb{N}$. For all $y_{k-1}, z_k \in \mathbb{R}^d$ and $A_k \geq 0$, the iterates of Algorithm 17 satisfy

$$\phi_{k+1} \leq \phi_k,$$

Algorithm 17 Information-theoretic exact method (ITEM)

Input: L-smooth μ-strongly (possibly with $\mu = 0$) convex function f and initial x_0.

Initialize $z_0 = x_0$ and $A_0 = 0$; $q = \mu/L$ (inverse condition ratio).
for $k = 0, \dots$ **do**
  $A_{k+1} = \frac{(1+q)A_k + 2\left(1+\sqrt{(1+A_k)(1+qA_k)}\right)}{(1-q)^2}$
  Set $\tau_k = 1 - \frac{A_k}{(1-q)A_{k+1}}$, and $\delta_k = \frac{1}{2}\frac{(1-q)^2 A_{k+1} - (1+q)A_k}{1+q+qA_k}$
  $y_k = x_k + \tau_k(z_k - x_k)$
  $x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k)$
  $z_{k+1} = (1 - q\delta_k)z_k + q\delta_k y_k - \frac{\delta_k}{L}\nabla f(y_k)$
end for

Output: Approximate solutions (y_k, x_{k+1}, z_{k+1}).

with

$$\begin{aligned}\phi_k \triangleq & A_k\Big[f(y_{k-1}) - f_\star - \frac{1}{2L}\|\nabla f(y_{k-1})\|_2^2 \\ & - \frac{\mu}{2(1-q)}\|y_{k-1} - \tfrac{1}{L}\nabla f(y_{k-1}) - x_\star\|_2^2\Big] \\ & + \frac{L+\mu A_k}{1-q}\|z_k - x_\star\|_2^2\end{aligned}$$

and $A_{k+1} = \frac{(1+q)A_k + 2\left(1+\sqrt{(1+A_k)(1+qA_k)}\right)}{(1-q)^2}$.

Proof. We first perform a weighted sum of two inequalities from Theorem 4.4.

- Smoothness and strong convexity between y_{k-1} and y_k with weight $\lambda_1 = A_k$:

$$\begin{aligned}f(y_{k-1}) \geq & f(y_k) + \langle \nabla f(y_k); y_{k-1} - y_k\rangle \\ & + \frac{1}{2L}\|\nabla f(y_k) - \nabla f(y_{k-1})\|_2^2 \\ & + \frac{\mu}{2(1-q)}\|y_k - y_{k-1} - \tfrac{1}{L}(\nabla f(y_k) - \nabla f(y_{k-1}))\|_2^2.\end{aligned}$$

- Smoothness and strong convexity of f between $x_\star$ and y_k with weight $\lambda_2 = A_{k+1} - A_k$:

$$\begin{aligned}f_\star \geq & f(y_k) + \langle \nabla f(y_k); x_\star - y_k\rangle + \frac{1}{2L}\|\nabla f(y_k)\|_2^2 \\ & + \frac{\mu}{2(1-q)}\|y_k - x_\star - \tfrac{1}{L}\nabla f(y_k)\|_2^2.\end{aligned}$$

By summing up and reorganizing these two inequalities (without substituting A_{k+1} by

its expression, for simplicity), we arrive at the following valid inequality:

$$\begin{aligned}0 \geq \lambda_1 \Big[& f(y_k) - f(y_{k-1}) + \langle \nabla f(y_k); y_{k-1} - y_k \rangle \\ & + \frac{1}{2L} \|\nabla f(y_k) - \nabla f(y_{k-1})\|_2^2 \\ & + \frac{\mu}{2(1-q)} \|y_k - y_{k-1} - \tfrac{1}{L}(\nabla f(y_k) - \nabla f(y_{k-1}))\|_2^2 \Big] \\ + \lambda_2 \Big[& f(y_k) - f_\star + \langle \nabla f(y_k); x_\star - y_k \rangle + \frac{1}{2L} \|\nabla f(y_k)\|_2^2 \\ & + \frac{\mu}{2(1-q)} \|y_k - x_\star - \tfrac{1}{L} \nabla f(y_k)\|_2^2 \Big].\end{aligned}$$

By substituting the expressions of y_k and z_{k+1} with

$$\begin{aligned} y_k &= (1 - \tau_k) \left(y_{k-1} - \tfrac{1}{L} \nabla f(y_{k-1}) \right) + \tau_k z_k \\ z_{k+1} &= (1 - q\delta_k) z_k + q\delta_k y_k - \tfrac{\delta_k}{L} \nabla f(y_k) \end{aligned}$$

(noting that this substitution is valid even for $k = 0$ since $A_0 = 0$ in that case and hence, $\tau_0 = 1$ and $y_0 = z_0$), the previous inequality can be reformulated exactly as

$$\begin{aligned}\phi_{k+1} \leq & \phi_k - \frac{LK_1}{1-q} P(A_{k+1}) \|z_k - x_\star\|_2^2 \\ & + \frac{K_2}{4L(1-q)} P(A_{k+1}) \\ & \times \|(1-q)A_{k+1} \nabla f(y_k) - \mu A_k (x_k - x_\star) + K_3 \mu (z_k - x_\star)\|_2^2,\end{aligned}$$

with the three parameters (which are well-defined given that $0 \leq \mu < L$ and $A_k, A_{k+1} \geq 0$)

$$\begin{aligned} K_1 &= \frac{q^2}{(1+q)^2 + (1-q)^2 q A_{k+1}} \\ K_2 &= \frac{(1+q)^2 + (1-q)^2 q A_{k+1}}{(1-q)^2 \left(1 + q + q A_k\right)^2 A_{k+1}^2} \\ K_3 &= (1+q) \frac{(1+q)A_k - (1-q)(2+qA_k) A_{k+1}}{(1+q)^2 + (1-q)^2 q A_{k+1}}, \end{aligned}$$

as well as

$$P(A_{k+1}) = (A_k - (1-q)A_{k+1})^2 - 4A_{k+1}(1 + qA_k).$$

To obtain the desired inequality, we select A_{k+1} such that $A_{k+1} \geq A_k$ and $P(A_{k+1}) = 0$, thereby demonstrating the claim $\phi_{k+1} \leq \phi_k$ and the choice of A_{k+1}. ■

The final bound for this method is obtained after the usual growth analysis of the sequence A_k, as follows. When $\mu = 0$, we have

$$A_{k+1} = 2 + A_k + 2\sqrt{1 + A_k} \geq 2 + A_k + 2\sqrt{A_k} \geq (1 + \sqrt{A_k})^2,$$

reaching $\sqrt{A_{k+1}} \geq 1 + \sqrt{A_k}$ and hence $\sqrt{A_k} \geq k$ and $A_k \geq k^2$. When $\mu > 0$, we can use an alternate bound:

$$\begin{aligned} A_{k+1} =& \frac{(1+q)A_k + 2\left(1 + \sqrt{(1+A_k)(1+qA_k)}\right)}{(1-q)^2} \\ \geq& \frac{(1+q)A_k + 2\sqrt{qA_k^2}}{(1-q)^2} = \frac{A_k}{(1-\sqrt{q})^2}, \end{aligned}$$

therefore achieving similar bounds as before. In this case, we only emphasize the convergence result for $\|z_N - x_\star\|$ since it corresponds to the lower complexity bound for smooth strongly convex minimization (provided below).

Corollary 4.17. Let $f \in \mathcal{F}_{\mu,L}$, and denote $q = \mu/L$. For any $x_0 = z_0 \in \mathbb{R}^d$ and $N \in \mathbb{N}$ with $N \geq 1$, the iterates of Algorithm 17 satisfy

$$\|z_N - x_\star\|_2^2 \leq \frac{1}{1+qA_N}\|z_0 - x_\star\|_2^2 \leq \frac{(1-\sqrt{q})^{2N}}{(1-\sqrt{q})^{2N} + q}\|z_0 - x_\star\|_2^2.$$

Proof. From Theorem 4.16, we get

$$\phi_N \leq \phi_{N-1} \leq \ldots \leq \phi_0 = \frac{L}{1-q}\|z_0 - x_\star\|_2^2.$$

From (4.4), we have that $\phi_N \geq \frac{L+A_N\mu}{1-q}\|z_N - x_\star\|_2^2$. It remains to use the bounds on A_N. That is, by using $A_1 = \frac{4}{(1-q)^2} = \frac{4}{(1+\sqrt{q})^2(1-\sqrt{q})^2} \geq (1-\sqrt{q})^{-2}$, we have $A_N \geq (1-\sqrt{q})^{-2N}$, which concludes the proof. ∎

Before concluding, we mention that the algorithm is non-improvable when minimizing large-scale smooth strongly convex functions in the following sense.

Theorem 4.18. (Drori and Taylor, 2021, Corollary 4) Let $0 \leq \mu < L < \infty$, $d, N \in \mathbb{N}$ with $d \geq 2N + 1$. For any black-box first-order method that performs at most N calls to the first-order oracle $(f(\cdot), \nabla f(\cdot))$, there exists a function $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ and $x_0 \in \mathbb{R}^d$ such that

$$\|x_N - x_\star\|_2^2 \geq \frac{1}{1+qA_N}\|x_0 - x_\star\|_2^2,$$

where $x_\star \in \operatorname{argmin}_x f(x)$, x_N is the output of the method under consideration, x_0 is its input, and A_N is defined as in Algorithm 17.

Remark 4.5. Just as for the optimized gradient method from Section 4.3, the ITEM might serve as a template for the design of other accelerated schemes. However, it has the same caveats as the optimized gradient method, which are also similar to those of the triple momentum method, presented in the next section. As emphasized in Section 4.3.3, it is unclear how to generalize the ITEM to broader classes of problems, e.g., problems involving constraints.

4.6.2 The Triple Momentum Method

The *triple momentum method* (TMM) is due to Van Scoy *et al.* (2017) and is reminiscent of Nesterov's method with constant momentum, provided as Algorithm 15. It corresponds to the asymptotic behavior of the information-theoretic exact method, just as Nesterov's accelerated method with constant momentum is the limit case of Nesterov's method (see Section 4.5.3) and as Polyak's heavy-ball is the limit case of Chebyshev's method (see Section 2.3.3). Indeed, considering Algorithm 17, one can explicitly compute

$$\lim_{A_k \to \infty} \frac{A_{k+1}}{A_k} = (1 - \sqrt{q})^{-2}$$

as well as

$$\lim_{A_k \to \infty} \tau_k = 1 - \frac{1 - \sqrt{q}}{1 + \sqrt{q}}, \quad \lim_{A_k \to \infty} \delta_k = \frac{1}{\sqrt{q}}.$$

Algorithm 18 Triple momentum method (TMM)

Input: L-smooth μ-strongly convex function f and initial point x_0.

1: **Initialize** $y_{-1} = z_0 = x_0$; $q = \mu/L$ (inverse condition ratio).
2: **for** $k = 0, \dots$ **do**
3: $\quad A_{k+1} = \frac{A_k}{(1-\sqrt{q})^2}$ {Only for the proof/relation to previous methods.}
4: $\quad y_k = \frac{1-\sqrt{q}}{1+\sqrt{q}} \left(y_{k-1} - \frac{1}{L} \nabla f(y_{k-1}) \right) + \left(1 - \frac{1-\sqrt{q}}{1+\sqrt{q}} \right) z_k$
5: $\quad z_{k+1} = \sqrt{q} \left(y_k - \frac{1}{\mu} \nabla f(y_k) \right) + (1 - \sqrt{q})\, z_k$
6: **end for**

Output: Approximate solutions (y_k, z_{k+1}).

As for the information-theoretic exact method, the known potential function for the triple momentum method relies on inequality (4.4), not only for its proof but to show that it is nonnegative, which follows from instantiating (4.4) at $y = x_\star$.

Theorem 4.19. Let f be an L-smooth μ-strongly convex function, $x_\star = \operatorname{argmin}_x f(x)$, and $k \in \mathbb{N}$. For any $x_k, z_k \in \mathbb{R}^d$, the iterates of Algorithm 18 satisfy

$$\begin{aligned} & f(y_k) - f_\star - \frac{1}{2L} \|\nabla f(y_k)\|_2^2 - \frac{\mu}{2(1-q)} \|y_k - x_\star - \tfrac{1}{L} \nabla f(y_k)\|_2^2 \\ & \qquad + \frac{\mu}{1-q} \|z_{k+1} - x_\star\|_2^2 \\ \leq & \rho^2 \Bigg(f(y_{k-1}) - f_\star - \frac{1}{2L} \|\nabla f(y_{k-1})\|_2^2 \\ & \qquad - \frac{\mu}{2(1-q)} \|y_{k-1} - x_\star - \tfrac{1}{L} \nabla f(y_{k-1})\|_2^2 + \frac{\mu}{1-q} \|z_k - x_\star\|_2^2 \Bigg), \end{aligned}$$

with $\rho = 1 - \sqrt{q}$.

Proof. For simplicity, we consider Algorithm 18 in the following form, parameterized by ρ

$$\begin{aligned} y_k &= \frac{\rho}{2-\rho}\left(y_{k-1} - \tfrac{1}{L}\nabla f(y_{k-1})\right) + \left(1 - \frac{\rho}{2-\rho}\right) z_k \\ z_{k+1} &= (1-\rho)\left(y_k - \tfrac{1}{\mu}\nabla f(y_k)\right) + \rho z_k. \end{aligned}$$

We combine the following two inequalities.

- Smoothness and strong convexity between y_{k-1} and y_k with weight $\lambda_1 = \rho^2$:

$$\begin{aligned} f(y_{k-1}) \geq & f(y_k) + \langle \nabla f(y_k); y_{k-1} - y_k \rangle \\ & + \frac{1}{2L}\|\nabla f(y_k) - \nabla f(y_{k-1})\|_2^2 \\ & + \frac{\mu}{2(1-q)}\|y_k - y_{k-1} - \tfrac{1}{L}(\nabla f(y_k) - \nabla f(y_{k-1}))\|_2^2. \end{aligned}$$

- Smoothness and strong convexity between $x_\star$ and y_k with weight $\lambda_2 = 1 - \rho^2$:

$$\begin{aligned} f_\star \geq & f(y_k) + \langle \nabla f(y_k); x_\star - y_k \rangle + \frac{1}{2L}\|\nabla f(y_k)\|_2^2 \\ & + \frac{\mu}{2(1-q)}\|y_k - x_\star - \tfrac{1}{L}\nabla f(y_k)\|_2^2, \end{aligned}$$

After some algebra, the weighted sum can be reformulated exactly as follows (it is simpler not to use the expression of ρ to verify this):

$$\begin{aligned} & f(y_k) - f_\star - \frac{1}{2L}\|\nabla f(y_k)\|_2^2 - \frac{\mu}{2(1-q)}\|y_k - x_\star - \tfrac{1}{L}\nabla f(y_k)\|_2^2 \\ & \quad + \frac{\mu}{1-q}\|z_{k+1} - x_\star\|_2^2 \\ \leq & \rho^2 \Bigg(f(y_{k-1}) - f_\star - \frac{1}{2L}\|\nabla f(y_{k-1})\|_2^2 \\ & \qquad - \frac{\mu}{2(1-q)}\|y_{k-1} - x_\star - \tfrac{1}{L}\nabla f(y_{k-1})\|_2^2 + \frac{\mu}{1-q}\|z_k - x_\star\|_2^2 \Bigg) \\ & \quad - \frac{q - (\rho-1)^2}{(\rho-2)\,(1-q)} \\ & \qquad \times \langle \nabla f(y_k); \rho(y_{k-1} - x_\star) - 2(\rho-1)(z_k - x_\star) - \tfrac{\rho}{L}\nabla f(y_{k-1}) \rangle \\ & \quad - \frac{q - (\rho-1)^2}{(1-q)\,\mu}\|\nabla f(y_k)\|_2^2. \end{aligned}$$

Using the expression $\rho = 1 - \sqrt{q}$, the last two terms cancel, and we arrive at the desired

result:

$$\begin{aligned} & f(y_k) - f_\star - \frac{1}{2L}\|\nabla f(y_k)\|_2^2 - \frac{\mu}{2(1-q)}\|y_k - x_\star - \tfrac{1}{L}\nabla f(y_k)\|_2^2 \\ & \quad + \frac{\mu}{1-q}\|z_{k+1} - x_\star\|_2^2 \\ \leq & \rho^2\Bigg(f(y_{k-1}) - f_\star - \frac{1}{2L}\|\nabla f(y_{k-1})\|_2^2 \\ & \quad - \frac{\mu}{2(1-q)}\|y_{k-1} - x_\star - \tfrac{1}{L}\nabla f(y_{k-1})\|_2^2 + \frac{\mu}{1-q}\|z_k - x_\star\|_2^2\Bigg). \quad \blacksquare \end{aligned}$$

Remark 4.6. The triple momentum method was proposed in (Van Scoy *et al.*, 2017), heavily relying on the control-theoretic framework developed by (Lessard *et al.*, 2016) (which is discussed in Appendix C). It was further studied in Cyrus *et al.* (2018) from a robust control perspective and by Zhou *et al.* (2020) as an accelerated method for a different objective. The triple momentum method can also be obtained as a time-independent optimized gradient method (Lessard and Seiler, 2020). Of course, all of the drawbacks of the OGM and ITEM also apply to the triple momentum method, so the same questions related to generalizations of this scheme remain open. Furthermore, this method is defined only for $\mu > 0$, like Nesterov's method with constant momentum.

4.6.3 Quadratic Averaging and Geometric Descent

Accelerated methods tailored for the strongly convex setting, such as Algorithms 15 and 18, make use of two kinds of step sizes to update the different sequences. First, they perform *small gradient steps* with the step size $1/L$. Such steps correspond to minimizing quadratic upper bounds (4.2). Second, they use so-called *large steps* $1/\mu$, which correspond to minimizing quadratic lower bounds (4.3). This algorithmic structure is provided with an interpretation which was further exploited by *geometric descent* (Bubeck *et al.*, 2015) and *quadratic averaging* (Drusvyatskiy *et al.*, 2018). These two methods produce the same sequences of iterates (see (Drusvyatskiy *et al.*, 2018, Theorem 4.5)), and we therefore take the stand of presenting them through the quadratic averaging viewpoint, which relates more directly to previous sections.

Drusvyatskiy *et al.* (2018) propose a method based on *quadratic averaging*. It is similar in shape to Algorithm 15 except that the last sequence z_k is explicitly computed as the minimum of a quadratic lower bound on the function. (The coefficients arising in the computation of z_k are dynamically selected to maximize this lower bound). To construct the new lower bound at iteration k, the algorithm combines the lower bound constructed at iteration $k-1$, whose minimum is achieved at z_k, with a new lower bound constructed using the strong convexity assumption along with the first-order information of the current iterate (more precisely this second lower bound on $f(x)$ is $f(y_k) + \langle \nabla f(y_k), x - y_k\rangle + \frac{\mu}{2}\|x - y_k\|_2^2$), whose minimum is $y_k - \frac{1}{\mu}\nabla f(y_k)$. Because of the specific shape of these two lower bounds, their convex combinations has a minimum that is the convex combination of their respective minima, with the same weights, thereby

motivating the update rule $z_k = \lambda(y_k - \frac{1}{\mu}\nabla f(y_k)) + (1-\lambda)z_{k-1}$, with dynamically chosen λ (to maximize the minimum value of the new under-approximation).

Alternatively, the sequence z_k can be interpreted in terms of a localization method for tracking $x_\star$, using intersections of balls. In this case, the sequence z_k corresponds to the centers of the balls containing $x_\star$. This alternate viewpoint is referred to as *geometric descent* (Bubeck *et al.*, 2015), and the λ are chosen to minimize the radius of the ball, centered at $z_k = \lambda(y_k - \frac{1}{\mu}\nabla f(y_k)) + (1-\lambda)z_{k-1}$ while ensuring that the new ball contains $x_\star$.

Geometric descent is detailed in (Bubeck *et al.*, 2015) and (Bubeck, 2015, Section 3.6.3). It has been extended to handle constraints (Chen *et al.*, 2017) and has been studied using the same Lyapunov function as that used in Theorem 4.14 (Karimi and Vavasis, 2017).

4.7 Practical Extensions

The goal of this section is to detail a few of the many extensions of Nesterov's accelerated methods. We see below that additional elements can be incorporated into the accelerated methods while maintaining the same proof structures. That is, we perform weighted sums involving the same inequalities that we used for the smooth (possibly strongly) convex functions and only introduce a few additional inequalities to account for the new elements.

We also seek to provide intuition, along with bibliographical pointers for going further. The following scenarios are particularly important in practice.

Constraints. In the presence of constraints or nonsmooth functions, a common approach is to resort to (proximal) splitting techniques. This idea is not recent; see, e.g., (Douglas and Rachford, 1956; Glowinski and Marroco, 1975; Lions and Mercier, 1979). However, it remains highly relevant in signal processing, computer vision, and statistical learning (Peyré, 2011; Parikh and Boyd, 2014; Chambolle and Pock, 2016; Fessler, 2020).

Adaptation. Problem constants, such as smoothness and strong convexity parameters, are generally unknown. Furthermore, their *local* values tend to be much more favorable than their, typically conservative, global values. In general, smoothness constants are estimated on the fly using backtracking line-search strategies; see, e.g. (Goldstein, 1962; Armijo, 1966; Nesterov, 1983). Strong convexity constants, or the more general Hölderian error bounds (see Section 6), on the other hand, are more difficult to estimate, and *restart* schemes are often used to adapt to these additional regularity properties; see, e.g. (Nesterov, 2013; Becker *et al.*, 2011; O'Donoghue and Candes, 2015; Roulet and d'Aspremont, 2017). Such schemes are the workhorse of Section 6.

Non-Euclidean settings. Although we only briefly mention this topic, accounting for the geometry of the problem at hand is generally key to obtaining good empirical per-

formance. In particular, optimization problems can often be formulated more naturally in a non-Euclidean space, with non-Euclidean norms producing better implicit models for the function. A popular method in this setting is commonly known as mirror descent (Nemirovsky and Yudin, 1983a)—see also (Ben-Tal and Nemirovsky, 2001; Juditsky and Nemirovsky, 2011a)—which we do not to detail at length here. Good surveys are provided by Beck and Teboulle (2003) and Bubeck (2015). However, we do describe an accelerated method in this setting in Section 4.7.4.

Numerical stability and monotone accelerated methods. Gradient descent guarantees the iterates to be monotonically good approximations of an optimal solution (i.e., $f(x_{k+1}) \leq f(x_k)$ for all k). This desirable property is generally not true for accelerated methods. Although the worst-case guarantees on $f(x_k) - f_\star$ are indeed monotonically decreasing functions of the iteration counter (such methods are sometimes referred to as *quasi-monotone methods* (Nesterov and Shikhman, 2015)), accelerated methods are in general not descent schemes. Monotonicity is a desirable feature for improving numerical stability of algorithms, and we show in Section 4.7.3 that simple modifications allows enforcing monotonicity of accelerated methods at low technical and computational cost. Albeit with a different presentation, such developments can be found in, e.g., (Tseng, 2008; Beck and Teboulle, 2009b). The technique is particularly simple to incorporate within the potential function-based analyses of this section.

4.7.1 Handling Nonsmooth Terms/Constraints

In this section, we consider the problem of minimizing a sum of two convex functions:

$$F_\star = \min_{x \in \mathbb{R}^d} \{F(x) \triangleq f(x) + h(x)\}, \tag{4.19}$$

where f is L-smooth and (possibly μ-strongly) convex and where h is convex, closed, and proper (CCP), which we denote by $f \in \mathcal{F}_{\mu,L}$ and $h \in \mathcal{F}_{0,\infty}$ (These technical conditions ensure that the proximal operator, defined hereafter, is well defined everywhere on $\mathbb{R}^d$. We refer to the clear introduction by Ryu and Boyd (2016) and the references therein for further details.) In addition, we assume a proximal operator of h to be readily available, so

$$\text{prox}_{\gamma h}(x) \triangleq \operatorname*{argmin}_y \{\gamma h(y) + \tfrac{1}{2}\|x - y\|_2^2\}, \tag{4.20}$$

can be evaluated efficiently. (Section 5 deals with some cases where this operator is approximated; see also the discussions in Section 4.9.) The proximal operator can be seen as an *implicit (sub)gradient* step on h, as dictated by the optimality conditions of the proximal operation

$$x_+ = \text{prox}_{\gamma h}(x) \Leftrightarrow x_+ = x - \gamma g_h(x_+) \text{ with } g_h(x_+) \in \partial h(x_+).$$

In particular, when $h(x)$ is the indicator function of a closed convex set Q, the proximal operation corresponds to the orthogonal projection onto Q. There are a few commonly

used functions for which the proximal operation has an analytical solution such as $h(x) = \|x\|_1$; see, for instance, the list provided by (Chierchia *et al.*, 2020). In the proofs below, we incorporate h using inequalities that characterize convexity, that is,

$$h(x) \geq h(y) + \langle g_h(y); x - y \rangle,$$

where $g_h(y) \in \partial h(y)$ is some subgradient of h at y. The proximal step (sometimes referred to as backward, or implicit, step) is a base algorithmic tool in the first-order optimization toolbox.

In this setting, classical methods for solving (4.19) involve a *forward-backward splitting* strategy (in other words, forward steps (a.k.a. gradient steps) on f and backward steps (a.k.a. proximal steps) on h), introduced by Passty (1979). This topic is addressed in many references, and we refer to (Parikh and Boyd, 2014; Ryu and Boyd, 2016) and the references therein for further details. In the context of accelerated methods, forward-backward splitting was introduced by Nesterov (2003; 2013) through the concept of *gradient mapping*; see also Tseng (2008) and Beck and Teboulle (2009a). Problem (4.19) is also sometimes referred to as the *composite convex optimization setting* (Nesterov, 2013). Depending on the assumptions made on f and h, there are alternate ways of solving this problem—for example, when the proximal operator is available for both, one can use the Douglas-Rachford splitting (Douglas and Rachford, 1956; Lions and Mercier, 1979). However, this is beyond the scope of this section and we refer to (Ryu and Boyd, 2016; Condat *et al.*, 2019) and the references therein for further discussions on this topic.

4.7.2 Adaptation to Unknown Regularity Parameters

In previous sections, we assumed f to be L-smooth and possibly μ-strongly convex. Moreover, in the previous algorithms, we explicitly used the values of both L and μ to design the methods. However, this is not a desirable feature. First, it means that we need to be able to estimate valid values for L and μ. Second, it means that the methods are not adaptive to potentially better (local) parameter values. That is, we do not benefit from the problems being simpler than specified, i.e., where the smallest valid L is much smaller than our estimate and/or the largest valid μ is much larger than our estimation. Furthermore, we want to benefit from the typically better local properties of the function at hand, along the path taken by the method, rather than relying on the global properties. The difference between local and global regularity properties is often significant, and adaptive methods often converge much faster in practice.

We discuss below how adaptation is implemented for the smoothness constant, using line-search techniques. However, it remains an open question whether strong convexity parameters can be efficiently estimated while maintaining reasonable worst-case guarantees and without resorting to restart schemes (i.e., outer iterations) (see Section 6).

To handle unknown parameters, the key is to examine the inequalities used in the proofs of the desired method. It turns out that smoothness is usually only used in inequalities between pairs of iterates, which means that these inequalities can be tested

at runtime, at each iteration. Therefore, for our guarantees to hold, we do not need the function to be L-smooth everywhere, but rather we only need a given inequality to hold for the value of L that we are using (where the smoothness of the function ensures that such an L exists). Conversely, strong convexity is typically only used in inequalities involving the optimal point (see, for example, the proof of Theorem 4.12), which we do not know a priori. As a result, these inequalities cannot be tested as the algorithm proceeds, which complicates the estimation of strong convexity while running the algorithm. Adaptation to strong convexity is therefore typically accomplished via the use of *restarts*.

These approaches are common, and are not new (Goldstein, 1962; Armijo, 1966); they were already used by Nesterov (1983). They were later adapted to the forward-backward setting in Nesterov (2013) and Beck and Teboulle (2009a) and have been further exploited to improve performance in various settings; see, e.g., (Scheinberg *et al.*, 2014). The topic is further discussed in the next section as well as in the notes and references provided in Section 4.9.

An Accelerated Forward-backward Methods with Backtracking

As discussed above, the smoothness constant L is used very sparsely in the proofs of both the gradient descent (Theorem 4.2 and Theorem 4.10) and the accelerated variants (Theorem 4.8, Theorem 4.12, and Theorem 4.14). Essentially, it is only used in three places: (i) to compute A_{k+1} (only when $\mu > 0$); (ii) to compute $x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k)$; and (iii) in the inequality

$$f(x_{k+1}) \leq f(y_k) + \langle \nabla f(y_k); x_{k+1} - y_k \rangle + \frac{L}{2}\|x_{k+1} - y_k\|_2^2. \tag{4.21}$$

(Recall that this is known as the *descent lemma* since by substituting the gradient step $x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k)$ it can be written as $f(x_{k+1}) \leq f(y_k) - \frac{1}{2L}\|\nabla f(y_k)\|_2^2$.) Other than L, (4.21) only contains information that *we observe*. Hence, we can simply *check* whether this inequality holds for a given estimate of L. When it does not hold, we simply increase the current approximation of L and then with this new estimate, recompute (i) A_{k+1} (necessary only if $\mu > 0$) and the corresponding y_k and (ii) x_{k+1}, using the new step size. We then check again whether (4.21) is satisfied. If (4.21) is satisfied, then we can proceed (because the potential inequality of the desired method is then verified—see, e.g., Theorem 4.2 or Theorem 4.10 for gradient descent, or Theorem 4.8, Theorem 4.12, or Theorem 4.14 for Nesterov's methods), and otherwise we continue increasing our approximation of L until the descent condition (4.21) is satisfied. Finally, to guarantee that we only perform a finite number of "wasted" gradient steps to estimate L, we need an appropriate rule for how to increase our approximation. It is common to simply multiply the current approximation by some constant $\alpha > 1$, thereby guaranteeing that at most $\lceil \log_\alpha \frac{L}{L_0} \rceil$ gradient steps, where L is the true smoothness constant and L_0 is our starting estimate, are wasted in the process. As we see below, both backtracking and nonsmooth

terms require proofs very similar to those presented above, and potential-based analyses are suitable.

We present two extensions of Nesterov's first method that can handle nonsmooth terms, and that have a backtracking procedure on the smoothness parameter. The first, the fast iterative shrinkage-thresholding algorithm (FISTA), is particularly popular, while the second resolves one potential issue that can arise in the original FISTA.

FISTA. The fast iterative shrinkage-thresholding algorithm, due to Beck and Teboulle (2009a), is a natural extension of Nesterov (1983) in its first form (see Algorithm 14), handling an additional nonsmooth term. In this section, we present a strongly convex variant of FISTA, provided as Algorithm 19. The proof contains the same ingredients as in the original work, and it can easily be compared to previous material.

Algorithm 19 Strongly convex FISTA, form I

Input: L-smooth μ-strongly (possibly with $\mu = 0$) convex function f, a convex function h with proximal operator available, an initial point x_0, and an initial estimate $L_0 > \mu$.

Initialize $z_0 = x_0$, $A_0 = 0$, and some $\alpha > 1$.
for $k = 0, \dots$ **do**
  $L_{k+1} = L_k$
  loop
    $q_{k+1} = \mu / L_{k+1}$
    $A_{k+1} = \frac{2A_k + 1 + \sqrt{4A_k + 4q_{k+1}A_k^2 + 1}}{2(1 - q_{k+1})}$
    set $\tau_k = \frac{(A_{k+1} - A_k)(1 + q_{k+1}A_k)}{A_{k+1} + 2q_{k+1}A_kA_{k+1} - q_{k+1}A_k^2}$ and $\delta_k = \frac{A_{k+1} - A_k}{1 + q_{k+1}A_{k+1}}$
    $y_k = x_k + \tau_k(z_k - x_k)$
    $x_{k+1} = \operatorname{prox}_{h/L_{k+1}}\left(y_k - \frac{1}{L_{k+1}}\nabla f(y_k)\right)$
    $z_{k+1} = (1 - q_{k+1}\delta_k)z_k + q_{k+1}\delta_k y_k + \delta_k\left(x_{k+1} - y_k\right)$
    if (4.21) holds **then**
      break {Iterates accepted; k will be incremented.}
    else
      $L_{k+1} = \alpha L_{k+1}$ {Iterates not accepted; recompute new L_{k+1}.}
    end if
  end loop
end for

Output: An approximate solution x_{k+1}.

The proof follows exactly the same steps as the proof of Theorem 4.12 (Nesterov's method for strongly convex functions), but it also accounts for the nonsmooth function h. (Observe that the potential is stated in terms of F and not f.) Two additional inequalities, involving the convexity of h between two different pairs of points, allow this nonsmooth term to be taken into account. In this case, f is assumed to be smooth and convex over $\mathbb{R}^d$ (i.e., it has full domain, $\mathbf{dom}\, f = \mathbb{R}^d$), and we are therefore allowed to evaluate gradients of f outside of domain of h.

Theorem 4.20. Let $f \in \mathcal{F}_{\mu,L}$ (with full domain: $\mathbf{dom}\, f = \mathbb{R}^d$); h be a closed convex proper function; $x_\star \in \operatorname{argmin}_x \{f(x)+h(x)\}$; and $k \in \mathbb{N}$. For any $x_k, z_k \in \mathbb{R}^d$ and $A_k \geq 0$, the iterates of Algorithm 19 that satisfy (4.21) also satisfy

$$\begin{aligned} A_{k+1}(F(x_{k+1}) - F_\star) + \frac{L_{k+1}+\mu A_{k+1}}{2}\|z_{k+1} - x_\star\|_2^2 \\ \leq A_k(F(x_k) - F_\star) + \frac{L_{k+1}+\mu A_k}{2}\|z_k - x_\star\|_2^2, \end{aligned}$$

with $A_{k+1} = \frac{2A_k+1+\sqrt{4A_k+4A_k^2 q_{k+1}+1}}{2(1-q_{k+1})}$ and $q_{k+1} = \mu/L_{k+1}$.

Proof. The proof consists of a weighted sum of the following inequalities.

- Strong convexity of f between $x_\star$ and y_k with weight $\lambda_1 = A_{k+1} - A_k$:

$$f(x_\star) \geq f(y_k) + \langle \nabla f(y_k); x_\star - y_k\rangle + \frac{\mu}{2}\|x_\star - y_k\|_2^2.$$

- Strong convexity of f between x_k and y_k with weight $\lambda_2 = A_k$:

$$f(x_k) \geq f(y_k) + \langle \nabla f(y_k); x_k - y_k\rangle.$$

- Smoothness of f between y_k and x_{k+1} (*descent lemma*) with weight $\lambda_3 = A_{k+1}$:

$$f(y_k) + \langle \nabla f(y_k); x_{k+1} - y_k\rangle + \frac{L_{k+1}}{2}\|x_{k+1} - y_k\|_2^2 \geq f(x_{k+1}).$$

- Convexity of h between $x_\star$ and x_{k+1} with weight $\lambda_4 = A_{k+1} - A_k$:

$$h(x_\star) \geq h(x_{k+1}) + \langle g_h(x_{k+1}); x_\star - x_{k+1}\rangle,$$

  with $g_h(x_{k+1}) \in \partial h(x_{k+1})$ and $x_{k+1} = y_k - \frac{1}{L_{k+1}}(\nabla f(y_k) + g_h(x_{k+1}))$.

- Convexity of h between x_k and x_{k+1} with weight $\lambda_5 = A_k$:

$$h(x_k) \geq h(x_{k+1}) + \langle g_h(x_{k+1}); x_k - x_{k+1}\rangle.$$

We get the following inequality

$$\begin{aligned} 0 \geq &\lambda_1[f(y_k) - f(x_\star) + \langle \nabla f(y_k); x_\star - y_k\rangle + \frac{\mu}{2}\|x_\star - y_k\|_2^2] \\ &+ \lambda_2[f(y_k) - f(x_k) + \langle \nabla f(y_k); x_k - y_k\rangle] \\ &+ \lambda_3[f(x_{k+1}) - (f(y_k) + \langle \nabla f(y_k); x_{k+1} - y_k\rangle \\ &\quad + \frac{L_{k+1}}{2}\|x_{k+1} - y_k\|_2^2)] \\ &+ \lambda_4[h(x_{k+1}) - h(x_\star) + \langle g_h(x_{k+1}); x_\star - x_{k+1}\rangle] \\ &+ \lambda_5[h(x_{k+1}) - h(x_k) + \langle g_h(x_{k+1}); x_k - x_{k+1}\rangle]. \end{aligned}$$

By substituting y_k, x_{k+1}, and z_{k+1} with

$$\begin{aligned} y_k &= x_k + \tau_k(z_k - x_k) \\ x_{k+1} &= y_k - \tfrac{1}{L_{k+1}}(\nabla f(y_k) + g_h(x_{k+1})) \\ z_{k+1} &= (1 - q_{k+1}\delta_k)z_k + q_{k+1}\delta_k y_k + \delta_k\left(x_{k+1} - y_k\right), \end{aligned}$$

the previous weighted sum can be reformulated exactly as

$$\begin{aligned} &A_{k+1}(f(x_{k+1}) + h(x_{k+1}) - f(x_\star) - h(x_\star)) \\ &\quad + \frac{L_{k+1} + \mu A_{k+1}}{2}\|z_{k+1} - x_\star\|_2^2 \\ &\leq A_k(f(x_k) + h(x_k) - f(x_\star) - h(x_\star)) + \frac{L_{k+1} + A_k\mu}{2}\|z_k - x_\star\|_2^2 \\ &\quad + \frac{(A_k - A_{k+1})^2 - A_{k+1} - q_{k+1}A_{k+1}^2}{1 + q_{k+1}A_{k+1}}\frac{1}{2L_{k+1}}\|\nabla f(y_k) + g_h(x_{k+1})\|_2^2 \\ &\quad - \frac{A_k^2(A_{k+1} - A_k)(1 + q_{k+1}A_k)(1 + q_{k+1}A_{k+1})}{\left(A_{k+1} + 2q_{k+1}A_kA_{k+1} - q_{k+1}A_k^2\right)^2}\frac{\mu}{2}\|x_k - z_k\|_2^2. \end{aligned}$$

Using $0 \leq q_{k+1} \leq 1$ and selecting A_{k+1} such that $A_{k+1} \geq A_k$ and

$$(A_k - A_{k+1})^2 - A_{k+1} - q_{k+1}A_{k+1}^2 = 0,$$

yields the desired result:

$$\begin{aligned} &A_{k+1}(f(x_{k+1}) + h(x_{k+1}) - f(x_\star) - h(x_\star)) \\ &\quad + \frac{L_{k+1} + \mu A_{k+1}}{2}\|z_{k+1} - x_\star\|_2^2 \\ &\leq A_k(f(x_k) + h(x_k) - f(x_\star) - h(x_\star)) + \frac{L_{k+1} + \mu A_k}{2}\|z_k - x_\star\|_2^2. \end{aligned}$$

■

Finally, we obtain a complexity guarantee by adapting the potential argument (4.5) and by noting that A_{k+1} is a decreasing function of L_{k+1} (whose maximal value is αL assuming $L_0 < L$ and is otherwise L_0). The growth rate of A_k in the smooth convex setting remains unchanged; see (4.14). However, when $L_0 < L$, its geometric grow rate might actually be slightly degraded to

$$A_{k+1} \geq \left(1 - \sqrt{\frac{\mu}{\alpha L}}\right)^{-1} A_k,$$

which remains better than the worst-case $(1 - \frac{\mu}{\alpha L})$ rate of gradient descent with backtracking, assuming in both cases that $L_0 < L$. When $L_0 > L$ the rates might respectively be degraded to $(1 - \sqrt{\frac{\mu}{L_0}})$ and $(1 - \frac{\mu}{L_0})$ instead.

Corollary 4.21. Let $f \in \mathcal{F}_{\mu,L}$ (with full domain: $\mathbf{dom}\, f = \mathbb{R}^d$); h be a closed convex proper function; and $x_\star \in \operatorname{argmin}_x \{F(x) \triangleq f(x) + h(x)\}$. For any $N \in \mathbb{N}$, $N \geq 1$, and $x_0 \in \mathbb{R}^d$, the output of Algorithm 19 satisfies

$$F(x_N) - F_\star \leq \min\left\{\frac{2}{N^2}, \left(1 - \sqrt{\frac{\mu}{\ell}}\right)^N\right\}\ell\|x_0 - x_\star\|_2^2,$$

with $\ell = \max\{\alpha L, L_0\}$.

Proof. We assume that $L > L_0$ since otherwise $f \in \mathcal{F}_{\mu,L_0}$ and the proof would directly follow the case without backtracking. Define

$$\phi_k \triangleq A_k(F(x_k) - F_\star) + \frac{L_k + \mu A_k}{2}\|z_k - x_\star\|_2^2.$$

Since $L_{k+1}/L_k \geq 1$, we have

$$\phi_{k+1} \leq A_k(F(x_k) - F_\star) + \frac{L_{k+1} + \mu A_k}{2}\|z_k - x_\star\|_2^2 \leq \frac{L_{k+1}}{L_k}\phi_k.$$

The chained potential argument (4.5) can then be adapted to obtain

$$A_N(F(x_N) - F_\star) \leq \phi_N \leq \frac{L_N}{L_{N-1}}\phi_{N-1} \leq \frac{L_N}{L_{N-2}}\phi_{N-2} \leq \ldots \leq \frac{L_N}{L_0}\phi_0,$$

where we used Theorem 4.20 and the fact that the output of the algorithm satisfies (4.21). Using $A_0 = 0$, we reach

$$F(x_N) - F_\star \leq \frac{L_N\|x_0 - x_\star\|_2^2}{2A_N}.$$

Using our previous bounds on A_N (noting that A_{k+1} is a decreasing function of L_{k+1}) in, e.g., Corollary 4.13, along with the fact that in the worst-case the estimated smoothness cannot be larger than the growth rate α times the true constant $L_N < \alpha L$ except if L_0 were already larger than the true L, in which case $L_N = L_0$. Therefore, we get $L_N \leq \ell = \max\{\alpha L, L_0\}$, yielding the desired result. ∎

Remark 4.7. There are two common variations on the backtracking strategy presented in this section. One can, for example, reset $L_{k+1} \leftarrow L_0$ (in line 3 of Algorithm 19) at each iteration, potentially using a total of $N\lceil\log_\alpha \frac{L}{L_0}\rceil$ additional gradient evaluations over all iterations. Another possibility is to pick some additional constant $0 < \beta < 1$ and to initiate $L_{k+1} \leftarrow \beta L_k$ (in line 3 of Algorithm 19). In the case $\beta = 1/\alpha$, this strategy potentially costs 1 additional gradient evaluation per iteration due to the backtracking strategy, thus potentially using a total of $N + \lceil\log_\alpha \frac{L}{L_0}\rceil$ additional gradient evaluations over all iterations.

Such *non-monotonic* estimations of L can be incorporated at a low additional technical cost. The corresponding methods and their analyses are essentially the same as those of this section; they are provided in Appendix B.3.1 and B.3.2 (see Algorithm 31 and Algorithm 32).

Remark 4.8. Variations on strongly convex extensions of FISTA, involving backtracking line-searches, can be found in, e.g., (Chambolle and Pock, 2016; Calatroni and Chambolle, 2019; Florea and Vorobyov, 2018; Florea and Vorobyov, 2020), together with practical improvements. The method presented in this section was designed for easy comparison with the previous material.

Another Accelerated Proximal Gradient Method

FISTA potentially evaluates gradients outside of the domain of h, and it therefore implicitly assumes that f is defined even outside this region. In many situations, this is not an issue, such as when f is quadratic. In this section, we instead assume that f is continuously differentiable and satisfies smoothness condition (4.22) only for all $x, y \in \mathbf{dom}\, h$.

Definition 4.2. Let $0 \leq \mu < L \leq +\infty$ and $C \subseteq \mathbb{R}^d$. A closed convex proper function $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is L-smooth and μ-strongly convex on C (written $f \in \mathcal{F}_{\mu,L}(C)$) if and only if

- (L-smoothness) there exists an open set C' such that $C \subseteq C'$ and f is continuously differentiable on C', and for all $x, y \in C$, it holds that
$$f(x) \leq f(y) + \langle \nabla f(y); x - y \rangle + \frac{L}{2}\|x - y\|_2^2; \tag{4.22}$$
- (μ-strong convexity) for all $x, y \in C$, it holds that
$$f(x) \geq f(y) + \langle g_f(y); x - y \rangle + \frac{\mu}{2}\|x - y\|_2^2, \tag{4.23}$$
where $g_f(y) \in \partial f(y)$ is a subgradient of f at y. (Note that $g_f(y) = \nabla f(y)$ when f is differentiable.)

By extension, $\mathcal{F}_{\mu,\infty}(C)$ denotes the set of closed μ-strongly convex proper functions whose domain contains C, and $\mathcal{F}_{0,\infty}$ denotes the set of closed convex proper functions.

There exist different ways of handling this situation. The method presented in this section relies on using the proximal operator on the sequence z_k and on formulating Nesterov's method in form III (see Algorithm 13). In this situation, assuming the initial point is feasible ($F(x_0) < \infty$) implies both the x_k and y_k are obtained from convex combinations of feasible points and hence are feasible.

A wide variety of accelerated methods exists; most variants solve the issue of FISTA using two proximal operations per iteration (on both of the sequences x_k and z_k). The variant in this section performs only one projection per iteration, while also fixing the infeasibility issue of y_k in FISTA. Variations on this theme can found in a number of references; see, for example, (Auslender and Teboulle, 2006, "Improved interior gradient algorithm"), (Tseng, 2008, Algorithm 1), or more recently (Gasnikov and Nesterov, 2018, "Method of similar triangles").

Theorem 4.22. Let $h \in \mathcal{F}_{0,\infty}$, $f \in \mathcal{F}_{\mu,L}(\mathbf{dom}\, h)$; $x_\star \in \operatorname{argmin}_x \{F(x) \triangleq f(x) + h(x)\}$; and $k \in \mathbb{N}$. For any $x_k, z_k \in \mathbb{R}^d$ and $A_k \geq 0$, the iterates of Algorithm 20 that satisfy (4.21) also satisfy
$$\begin{aligned} A_{k+1}(F(x_{k+1}) - F_\star) + \frac{L_{k+1} + \mu A_{k+1}}{2}\|z_{k+1} - x_\star\|_2^2 \\ \leq A_k(F(x_k) - F_\star) + \frac{L_{k+1} + \mu A_k}{2}\|z_k - x_\star\|_2^2, \end{aligned}$$

Algorithm 20 A proximal accelerated gradient method

Input: $h \in \mathcal{F}_{0,\infty}$ with proximal operator available, $f \in \mathcal{F}_{\mu,L}(\mathbf{dom}\, h)$, an initial point $x_0 \in \mathbf{dom}\, h$, and an initial estimate $L_0 > \mu$.

Initialize $z_0 = x_0$, $A_0 = 0$, and some $\alpha > 1$.
for $k = 0, \ldots$ **do**
  $L_{k+1} = L_k$
  loop
    $q_{k+1} = \mu / L_{k+1}$
    $A_{k+1} = \frac{2A_k + 1 + \sqrt{4A_k + 4q_{k+1}A_k^2 + 1}}{2(1 - q_{k+1})}$
    set $\tau_k = \frac{(A_{k+1} - A_k)(1 + q_{k+1}A_k)}{A_{k+1} + 2q_{k+1}A_kA_{k+1} - q_{k+1}A_k^2}$ and $\delta_k = \frac{A_{k+1} - A_k}{1 + q_{k+1}A_{k+1}}$
    $y_k = x_k + \tau_k(z_k - x_k)$
    $z_{k+1} = \mathrm{prox}_{\delta_k h / L_{k+1}}\left((1 - q_{k+1}\delta_k)z_k + q_{k+1}\delta_k y_k - \frac{\delta_k}{L_{k+1}}\nabla f(y_k)\right)$
    $x_{k+1} = \frac{A_k}{A_{k+1}}x_k + (1 - \frac{A_k}{A_{k+1}})z_{k+1}$
    if (4.21) holds **then**
      break {Iterates accepted; k will be incremented.}
    else
      $L_{k+1} = \alpha L_{k+1}$ {Iterates not accepted; recompute new L_{k+1}}
    end if
  end loop
end for

Output: Approximate solution x_{k+1}.

with $A_{k+1} = \frac{2A_k + 1 + \sqrt{4A_k + 4A_k^2 q_{k+1} + 1}}{2(1 - q_{k+1})}$ and $q_{k+1} = \mu / L_{k+1}$.

Proof. First, z_k is in $\mathbf{dom}\, h$ by construction—it is the output of the proximal/projection step. Furthermore, we have $0 \leq \frac{A_k}{A_{k+1}} \leq 1$ given that $A_{k+1} \geq A_k \geq 0$. A direct consequence is that since $z_0 = x_0 \in \mathbf{dom}\, h$, all subsequent $\{y_k\}$ and $\{x_k\}$ are also in $\mathbf{dom}\, h$ (as they are obtained from convex combinations of feasible points).

The rest of the proof consists of a weighted sum of the following inequalities (which are valid due to the feasibility of the iterates).

- Strong convexity of f between $x_\star$ and y_k with weight $\lambda_1 = A_{k+1} - A_k$:
$$f(x_\star) \geq f(y_k) + \langle \nabla f(y_k); x_\star - y_k \rangle + \frac{\mu}{2}\|x_\star - y_k\|_2^2.$$
- Convexity of f between x_k and y_k with weight $\lambda_2 = A_k$:
$$f(x_k) \geq f(y_k) + \langle \nabla f(y_k); x_k - y_k \rangle.$$
- Smoothness of f between y_k and x_{k+1} (*descent lemma*) with weight $\lambda_3 = A_{k+1}$:
$$f(y_k) + \langle \nabla f(y_k); x_{k+1} - y_k \rangle + \frac{L_{k+1}}{2}\|x_{k+1} - y_k\|_2^2 \geq f(x_{k+1}).$$

- Convexity of h between $x_\star$ and z_{k+1} with weight $\lambda_4 = A_{k+1} - A_k$:
$$h(x_\star) \geq h(z_{k+1}) + \langle g_h(z_{k+1}); x_\star - z_{k+1}\rangle,$$
with $g_h(z_{k+1}) \in \partial h(z_{k+1})$ and $z_{k+1} = (1-q\delta_k)z_k + q\delta_k y_k - \frac{\delta_k}{L_{k+1}}(\nabla f(y_k) + g_h(z_{k+1}))$.
- Convexity of h between x_k and x_{k+1} with weight $\lambda_5 = A_k$:
$$h(x_k) \geq h(x_{k+1}) + \langle g_h(x_{k+1}); x_k - x_{k+1}\rangle,$$
with $g_h(x_{k+1}) \in \partial h(x_{k+1})$.
- Convexity of h between z_{k+1} and x_{k+1} with weight $\lambda_6 = A_{k+1} - A_k$:
$$h(z_{k+1}) \geq h(x_{k+1}) + \langle g_h(x_{k+1}); z_{k+1} - x_{k+1}\rangle.$$

We this obtain the following inequality:

$$\begin{aligned}
0 \geq & \lambda_1[f(y_k) - f(x_\star) + \langle \nabla f(y_k); x_\star - y_k\rangle + \frac{\mu}{2}\|x_\star - y_k\|_2^2] \\
& + \lambda_2[f(y_k) - f(x_k) + \langle \nabla f(y_k); x_k - y_k\rangle] \\
& + \lambda_3[f(x_{k+1}) - (f(y_k) + \langle \nabla f(y_k); x_{k+1} - y_k\rangle \\
& + \frac{L_{k+1}}{2}\|x_{k+1} - y_k\|_2^2)] \\
& + \lambda_4[h(z_{k+1}) - h(x_\star) + \langle g_h(z_{k+1}); x_\star - z_{k+1}\rangle] \\
& + \lambda_5[h(x_{k+1}) - h(x_k) + \langle g_h(x_{k+1}); x_k - x_{k+1}\rangle] \\
& + \lambda_6[h(x_{k+1}) - h(z_{k+1}) + \langle g_h(x_{k+1}); z_{k+1} - x_{k+1}\rangle].
\end{aligned}$$

By substituting y_k, z_{k+1}, and x_{k+1} with

$$\begin{aligned}
y_k &= x_k + \tau_k(z_k - x_k) \\
z_{k+1} &= (1 - q_{k+1}\delta_k)z_k + q_{k+1}\delta_k y_k - \frac{\delta_k}{L_{k+1}}(\nabla f(y_k) + g_h(z_{k+1})) \\
x_{k+1} &= \frac{A_k}{A_{k+1}}x_k + \left(1 - \frac{A_k}{A_{k+1}}\right) z_{k+1},
\end{aligned}$$

we reformulate the previous inequality as

$$\begin{aligned}
& A_{k+1}(f(x_{k+1}) + h(x_{k+1}) - f(x_\star) - h(x_\star)) \\
& \quad + \frac{L_{k+1} + A_{k+1}\mu}{2}\|z_{k+1} - x_\star\|_2^2 \\
& \leq A_k(f(x_k) + h(x_k) - f(x_\star) - h(x_\star)) + \frac{L_{k+1} + A_k\mu}{2}\|z_k - x_\star\|_2^2 \\
& \quad + \frac{(A_k - A_{k+1})^2\left((A_k - A_{k+1})^2 - A_{k+1} - q_{k+1}A_{k+1}^2\right)}{A_{k+1}(1 + q_{k+1}A_{k+1})^2} \\
& \qquad \times \frac{1}{2L_{k+1}}\|\nabla f(y_k) + g_h(z_{k+1})\|_2^2 \\
& \quad - \frac{A_k^2(A_{k+1} - A_k)(1 + q_{k+1}A_k)(1 + q_{k+1}A_{k+1})}{\left(A_{k+1} + 2q_{k+1}A_kA_{k+1} - q_{k+1}A_k^2\right)^2}\frac{\mu}{2}\|x_k - z_k\|_2^2.
\end{aligned}$$

Then selecting $A_{k+1} \geq A_k$ such that

$$(A_k - A_{k+1})^2 - A_{k+1} - q_{k+1}A_{k+1}^2 = 0,$$

yields the desired result:

$$\begin{aligned} &A_{k+1}(f(x_{k+1}) + h(x_{k+1}) - f(x_\star) - h(x_\star)) \\ &\quad + \frac{L_{k+1} + \mu A_{k+1}}{2}\|z_{k+1} - x_\star\|_2^2 \\ &\leq A_k(f(x_k) + h(x_k) - f(x_\star) - h(x_\star)) + \frac{L_{k+1} + \mu A_k}{2}\|z_k - x_\star\|_2^2. \end{aligned}$$ ■

We have the following corollary.

Corollary 4.23. Let $h \in \mathcal{F}_{0,\infty}$, $f \in \mathcal{F}_{\mu,L}(\mathbf{dom}\, h)$, and $x_\star \in \operatorname{argmin}_x \{F(x) \triangleq f(x)+h(x)\}$. For any $N \in \mathbb{N}$, $N \geq 1$, and $x_0 \in \mathbb{R}^d$, the output of Algorithm 20 satisfies

$$F(x_N) - F_\star \leq \min\left\{\frac{2}{N^2}, \left(1 - \sqrt{\frac{\mu}{\ell}}\right)^{-N}\right\}\ell\|x_0 - x_\star\|_2^2,$$

with $\ell = \max\{\alpha L, L_0\}$.

Proof. The proof follows the same arguments as those for Corollary 4.21, using the potential from Theorem 4.22 with the fact the output of the algorithm satisfies (4.21). ■

Remark 4.9. In this section, we introduced backtracking techniques by examining how the inequalities are used in previous proofs. In particular, because smoothness is used only through the *descent lemma* in which the only *unknown* value is L, one can simply check this inequality at runtime. Another way to exploit the observation of which inequalities are needed in a proof is to identify *minimal* assumptions on the class of functions under which it is possible to prove accelerated rates; this topic is explored by, e.g., Necoara *et al.* (2019) and Hinder *et al.* (2020). More generally the same question holds for the ability to prove the convergence rates of simpler methods, such as gradient descent (Bolte *et al.*, 2017; Necoara *et al.*, 2019).

4.7.3 Monotone Accelerated Methods

As emphasized in previous sections, accelerated methods are *quasi-monotone*, meaning that their worst-case guarantees are decreasing functions of the number of iteration. However, they are generally not monotone, as the function values are not guaranteed to be improved from iteration to iteration.

In this section, we introduce a simple trick for making common accelerated methods monotone. The technique stems from a simple observation that the iterates $\{x_k\}_k$ can be slightly changed while maintaining the worst-case guarantees (see, e.g., (Tseng, 2008)). That is, we introduce an additional sequence, denoted by $\{\tilde{x}_k\}_k$ for which $\{F(\tilde{x}_k)\}_k$ is monotonically decreasing. For instance, for all problems on which $\{F(x_k)\}_k$ is already monotonically decreasing, we have $x_k = \tilde{x}_k$ for all k.

As it is, the trick does not apply to the optimized gradient method (Algorithm 9), the information theoretic exact method (Algorithm 17) and the triple momentum method (Algorithm 18) due to the slightly different structure of their potential functions. However, this trick is valid for all other methods presented in this section, as well as those presented in Appendix B and the proximal accelerated methods from Section 5.

Algorithm 21 Wrapper: monotone accelerated methods

Input: Pick an algorithm $\mathcal{A}$ among Algorithms {11, 12, 13, 14, 15, 16, 19, 20, 28, 29, 31, 32} and use the same inputs the same associated problem inputs, including an initial guess $x_0 \in \mathbb{R}^d$.

1: **Initialize** Execute initialization step from the chosen algorithm, as well as $\tilde{x}_0 = x_0$.
2: **for** $k = 0, \ldots$ **do**
3: $\quad x_k = \tilde{x}_k$
4: $\quad$ Execute one iteration of the chosen algorithm (one update for each sequence).
5: $\quad \tilde{x}_{k+1} = \operatorname{argmin}_x \{F(x) : x \in \{x_{k+1}, \tilde{x}_k\}\}$
6: **end for**

Output: Approximate solution x_{k+1}.

The following fact summarizes the result for this method. In a nutshell, the desired result (i.e., same worst-case guarantees as previous algorithms while having a monotone sequence $\{F(\tilde{x}_k)\}_k$) is achieved due to two fact. First, the sequence $\{\tilde{x}_k\}_k$ is constructed for satisfying $F(\tilde{x}_{k+1}) \leq F(x_{k+1})$, and hence the potential function analyses of all the algorithmic schemes is preserved. Secondly, the sequence is built for satisfying $F(\tilde{x}_{k+1}) \leq F(\tilde{x_k})$ and hence it is monotonically decreasing.

Theorem 4.24. Let $\mathcal{A}$ be an algorithm in {11, 12, 13, 14, 15, 16, 19, 20, 28, 29, 31, 32}, let $F \in \mathcal{F}_{0,\infty}$ be the convex function on which this algorithm is applied (possibly $F \triangleq f + h$ with f and h satisfying the input requirements under which $\mathcal{A}$ operates), and $x_0 \in \mathbb{R}^d$. Further let $\{\tilde{x}_k\}_k$ be the sequence generated by Algorithm 21 with $\mathcal{A}$ on F and x_0. It follows that $F(\tilde{x}_N)$ satisfies the same worst-case guarantee as that of $\mathcal{A}$:

$$F(\tilde{x}_N) - F_\star \leq \frac{L\|x_0 - x_\star\|_2^2}{2A_N},$$

with A_N being defined in $\mathcal{A}$. In addition, the sequence $\{F(\tilde{x}_k)\}_k$ is monotonically decreasing.

Proof. The proof follows from the fact that worst-case guarantees for all algorithms under consideration rely on the same potential function:

$$\phi_k \triangleq A_k(F(x_k) - F_\star) + \frac{L + \mu A_k}{2}\|z_k - x_\star\|_2^2.$$

Defining the alternate potential $\tilde{\phi}_k$ as

$$\tilde{\phi}_k \triangleq A_k(F(\tilde{x}_k) - F_\star) + \frac{L + \mu A_k}{2}\|z_k - x_\star\|_2^2,$$

it follows from $F(\tilde{x}_k) \leq F(x_k)$ and $A_k \geq 0$ that

$$\tilde{\phi}_k \leq \phi_k$$

for all $k \geq 0$. Furthermore, it follows from the fact that

$$\tilde{\phi}_{k+1} \leq \phi_{k+1} \leq \phi_k$$

for all $x_k \in \mathbb{R}^d$ that we can choose $x_k = \tilde{x}_k$, thereby reaching $\tilde{\phi}_{k+1} \leq \tilde{\phi}_k$. Therefore, it holds that

$$A_N(F(\tilde{x}_N) - F_\star) \leq \tilde{\phi}_N \leq \tilde{\phi}_0 = \frac{L\|x_0 - x_\star\|_2^2}{2}.$$

Hence, the same worst-case guarantees are achieved.

Finally, monotonicity is obtained by construction. That is, the sequence satisfies $F(\tilde{x}_{k+1}) \leq F(\tilde{x}_k)$ (for all $k \geq 0$) and hence is monotonically decreasing. ■

4.7.4 Beyond Euclidean Geometries using Mirror Maps

In this section, we put ourselves in a slightly different scenario, often referred to as the *non-Euclidean setting* or the *mirror descent setup*. We consider the convex minimization problem:

$$F_\star = \min_{x \in \mathbb{R}^d} \{F(x) \triangleq f(x) + h(x)\}, \tag{4.24}$$

with $h, f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ closed, convex, and proper. Furthermore, we assume f to be differentiable and to have Lipschitz gradients with respect to some (possibly non-Euclidean) norm $\|.\|$. That is, with the corresponding dual norm denoted by $\|s\|_* = \sup_x\{\langle s; x\rangle : \|x\| \leq 1\}$, we require

$$\|\nabla f(x) - \nabla f(y)\|_* \leq L\|x - y\|$$

for all $x, y \in \mathbf{dom}\, h$. In this setting, inequality (4.22) also holds (see Appendix A.2), and we, perhaps abusively, also denote $f \in \mathcal{F}_{0,L}(\mathbf{dom}\, h)$.

To solve (4.24), we define a few additional ingredients. First, we pick a 1-strongly convex, closed proper function $w : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ such that $\mathbf{dom}\, h \subseteq \mathbf{dom}\, w$. (Recall that by assumption on h, $\mathbf{dom}\, h \neq \emptyset$, and therefore $\mathbf{dom}\, w \neq \emptyset$.) These assumptions ensure that the proximal operations below are well defined; w is commonly referred to as the *distance generating function*. Under additional technical assumptions, w could be chosen as a strictly convex function instead, and similar algorithms can be used, but we focus on the strongly convex case here.

Finally, pick $g_w(y) \in \partial w(y)$, and define the Bregman divergence generated by w as

$$D_w(x; y) = w(x) - w(y) - \langle g_w(y); x - y\rangle, \tag{4.25}$$

which we use below as a notion of distance to generalize the previous proximal operator (4.20). Note that the Bregman divergence $D_w(.;.)$ generated by any subgradient of w at z_k is considered valid here.

The base ingredient we use to solve (4.24) is the Bregman proximal gradient step, with step size $\frac{a_k}{L}$:

$$z_{k+1} = \operatorname*{argmin}_y \left\{ \frac{a_k}{L} \left(h(y) + \langle \nabla f(y_k); y - y_k \rangle \right) + D_w(y; z_k) \right\}, \tag{4.26}$$

which corresponds to the usual Euclidean proximal gradient step when $w(x) = \frac{1}{2}\|x\|_2^2$. Under previous assumptions, (4.26) is well defined and we can explicitly write

$$g_w(z_{k+1}) \ni g_w(z_k) - \frac{a_k}{L} \left(\nabla f(y_k) + g_h(z_{k+1}) \right),$$

with some $g_w(z_{k+1}) \in \partial w(z_{k+1})$, $g_w(z_k) \in \partial w(z_k)$, and $g_h(z_{k+1}) \in \partial h(z_{k+1})$.

Under this construction, one can rely on (4.26) to solve (4.24) using Algorithm 22. Note that when w is differentiable (which is usually the case but which necessitates further discussions when requiring w to be closed, convex, and proper), we often refer to ∇w as a *mirror map*. This refers to a bijective mapping due to the strong convexity and differentiability of w. In this case, the iterations can be described as

$$\nabla w(z_{k+1}) = \nabla w(z_k) - \frac{a_k}{L} \left(\nabla f(y_k) + g_h(z_{k+1}) \right).$$

Algorithm 22 Proximal accelerated Bregman gradient method

Input: $h \in \mathcal{F}_{0,\infty}$, $f \in \mathcal{F}_{0,L}(\mathbf{dom}\, h)$, $w \in \mathcal{F}_{1,\infty}$ with $\mathbf{dom}\, h \subseteq \mathbf{dom}\, w$, and $x_0 \in \mathbf{dom}\, h$ (such that $\partial w(z_0) \neq \emptyset$).

1: **Initialize** $z_0 = x_0$ and $A_0 = 0$.
2: **for** $k = 0, \ldots$ **do**
3: $\quad a_k = \frac{1+\sqrt{4A_k+1}}{2}$
4: $\quad A_{k+1} = A_k + a_k$
5: $\quad y_k = \frac{A_k}{A_{k+1}} x_k + \left(1 - \frac{A_k}{A_{k+1}}\right) z_k$
6: $\quad z_{k+1} = \operatorname{argmin}_y \left\{ \frac{a_k}{L} \left(h(y) + \langle \nabla f(y_k); y - y_k \rangle \right) + D_w(y; z_k) \right\}$
7: $\quad x_{k+1} = \frac{A_k}{A_{k+1}} x_k + \left(1 - \frac{A_k}{A_{k+1}}\right) z_{k+1}$
8: **end for**

Output: Approximate solution x_{k+1}.

Theorem 4.25 provides a convergence guarantee for Algorithm 22 by a potential argument.

Theorem 4.25. Let $h \in \mathcal{F}_{0,\infty}$, $f \in \mathcal{F}_{0,L}(\mathbf{dom}\, h)$, $w \in \mathcal{F}_{1,\infty}$ with $\mathbf{dom}\, h \subseteq \mathbf{dom}\, w$, $x_\star \in \operatorname{argmin}_x \{F(x) \triangleq f(x) + h(x)\}$, and $k \in \mathbb{N}$. For any $x_k, z_k \in \mathbf{dom}\, h$ such that $\partial w(z_k) \neq \emptyset$ and $A_k \geq 0$; the iterates of Algorithm 22 satisfy

$$\begin{aligned} A_{k+1}(F(x_{k+1}) - F_\star) + L D_w(x_\star; z_{k+1}) \\ \leq A_k(F(x_k) - F_\star) + L D_w(x_\star; z_k), \end{aligned}$$

where $A_{k+1} = A_k + \frac{1+\sqrt{4A_k+1}}{2}$ and $D_w(.;.)$ is a Bregman divergence (4.25) with respect to w. Furthermore, $\partial w(z_{k+1}) \neq \emptyset$.

Proof. First, z_k is feasible, i.e., $z_k \in \mathbf{dom}\, h$, by construction. Indeed, z_0 is feasible by assumption, and the following iterates z_k ($k > 0$) are obtained after proximal steps; hence, $\partial h(z_k) \neq \emptyset$, and therefore, $z_k \in \mathbf{dom}\, h$.

Second, it can be directly verified that $0 \leq \frac{A_k}{A_{k+1}} \leq 1$ given that $A_{k+1} \geq A_k \geq 0$. It follows from $z_0 = x_0 \in \mathbf{dom}\, h$ that the elements of $\{y_k\}$ and $\{x_k\}$ are obtained as convex combinations of elements of $\{z_k\}$. Hence, the sequences $\{y_k\}$ and $\{x_k\}$ are also in $\mathbf{dom}\, h$, which is convex. The rest of the proof consists of a weighted sum of the following inequalities.

- Convexity of f between $x_\star$ and y_k with weight $\lambda_1 = A_{k+1} - A_k$:

$$f(x_\star) \geq f(y_k) + \langle \nabla f(y_k); x_\star - y_k \rangle.$$

- Convexity of f between x_k and y_k with weight $\lambda_2 = A_k$:

$$f(x_k) \geq f(y_k) + \langle \nabla f(y_k); x_k - y_k \rangle.$$

- Smoothness of f between y_k and x_{k+1} (*descent lemma*) with weight $\lambda_3 = A_{k+1}$:

$$f(y_k) + \langle \nabla f(y_k); x_{k+1} - y_k \rangle + \frac{L}{2}\|x_{k+1} - y_k\|^2 \geq f(x_{k+1}).$$

- Convexity of h between $x_\star$ and z_{k+1} with weight $\lambda_4 = A_{k+1} - A_k$:

$$h(x_\star) \geq h(z_{k+1}) + \langle g_h(z_{k+1}); x_\star - z_{k+1} \rangle,$$

  with $g_h(z_{k+1}) \in \partial h(z_{k+1})$.

- Convexity of h between x_k and x_{k+1} with weight $\lambda_5 = A_k$:

$$h(x_k) \geq h(x_{k+1}) + \langle g_h(x_{k+1}); x_k - x_{k+1} \rangle,$$

  with $g_h(x_{k+1}) \in \partial h(x_{k+1})$.

- Convexity of h between z_{k+1} and x_{k+1} with weight $\lambda_6 = A_{k+1} - A_k$

$$h(z_{k+1}) \geq h(x_{k+1}) + \langle g_h(x_{k+1}); z_{k+1} - x_{k+1} \rangle.$$

- Strong convexity of w between z_{k+1} and z_k with weight $\lambda_7 = L$

$$w(z_{k+1}) \geq w(z_k) + \langle g_w(z_k); z_{k+1} - z_k \rangle + \frac{1}{2}\|z_k - z_{k+1}\|^2,$$

  with $g_w(z_k) \in \partial w(z_k)$.

We thus obtain the following inequality:

$$\begin{aligned}0 \geq &\lambda_1[f(y_k) - f_\star + \langle \nabla f(y_k); x_\star - y_k\rangle] \\ &+ \lambda_2[f(y_k) - f(x_k) + \langle \nabla f(y_k); x_k - y_k\rangle] \\ &+ \lambda_3[f(x_{k+1}) - (f(y_k) + \langle \nabla f(y_k); x_{k+1} - y_k\rangle + \frac{L}{2}\|x_{k+1} - y_k\|^2)] \\ &+ \lambda_4[h(z_{k+1}) - h(x_\star) + \langle g_h(z_{k+1}); x_\star - z_{k+1}\rangle] \\ &+ \lambda_5[h(x_{k+1}) - h(x_k) + \langle g_h(x_{k+1}); x_k - x_{k+1}\rangle] \\ &+ \lambda_6[h(x_{k+1}) - h(z_{k+1}) + \langle g_h(x_{k+1}); z_{k+1} - x_{k+1}\rangle] \\ &+ \lambda_7[w(z_k) - w(z_{k+1}) + \langle g_w(z_k); z_{k+1} - z_k\rangle + \frac{1}{2}\|z_k - z_{k+1}\|^2].\end{aligned}$$

Now, by substituting (using $g_w(z_{k+1}) \in \partial w(z_{k+1})$)

$$\begin{aligned}y_k &= \frac{A_k}{A_{k+1}} x_k + \left(1 - \frac{A_k}{A_{k+1}}\right) z_k \\ g_w(z_{k+1}) &= g_w(z_k) - \frac{A_{k+1} - A_k}{L}\left(\nabla f(y_k) + g_h(z_{k+1})\right) \\ x_{k+1} &= \frac{A_k}{A_{k+1}} x_k + \left(1 - \frac{A_k}{A_{k+1}}\right) z_{k+1},\end{aligned}$$

we obtain exactly $\|x_{k+1} - y_k\|^2 = \frac{(A_{k+1}-A_k)^2}{A_{k+1}^2}\|z_k - z_{k+1}\|^2$. The weighted sum can then be reformulated exactly as

$$\begin{aligned}&A_{k+1}(F(x_{k+1}) - F_\star) + LD_w(x_\star, z_{k+1}) \\ &\quad \leq A_k(F(x_k) - F_\star) + LD_w(x_\star, z_k) \\ &\qquad + \frac{(A_k - A_{k+1})^2 - A_{k+1}}{A_{k+1}} \frac{L}{2}\|z_k - z_{k+1}\|^2,\end{aligned}$$

and we obtain the desired inequality from selecting A_{k+1} that satisfies $A_{k+1} \geq A_k$ and

$$(A_k - A_{k+1})^2 - A_{k+1} = 0. \quad \blacksquare$$

We conclude this section by providing a final corollary to describe the worst-case performance of the method.

Corollary 4.26. Let $h \in \mathcal{F}_{0,\infty}$, $f \in \mathcal{F}_{0,L}(\mathbf{dom}\, h)$, $w \in \mathcal{F}_{1,\infty}$ with $\mathbf{dom}\, h \subseteq \mathbf{dom}\, w$, and $x_\star \in \operatorname{argmin}_x\{F(x) \triangleq f(x) + h(x)\}$. For any $x_0 \in \mathbf{dom}\, h$ such that $\partial w(x_0) \neq \emptyset$ and any $N \in \mathbb{N}$; the iterates of Algorithm 22 satisfy

$$F(x_N) - F_\star \leq \frac{LD_w(x_\star; z_0)}{A_N} \leq \frac{4LD_w(x_\star; z_0)}{N^2}.$$

Proof. The claim directly follows from previous arguments using the potential $\phi_k \triangleq A_k(F(x_k) - F_\star) + LD_w(x_\star; z_k)$, along with $A_N \geq \frac{N^2}{4}$ from (4.14). $\blacksquare$

Remark 4.10. *Bregman* first-order methods are often split into two different families: *mirror descent* and *dual averaging* (which we did not explicitly mention here), for which we refer the reader to the discussions in (Bubeck, 2015, Chapter 4). The method presented in this section is essentially a case of (Tseng, 2008, Algorithm 1), and it corresponds to (Auslender and Teboulle, 2006, "Improved interior gradient algorithm") when the norm is Euclidean. It is also similar to (Tseng, 2008, Algorithm 3) and to (Gasnikov and Nesterov, 2018, "Method of similar triangles"), which are "dual-averaging" versions of the same algorithm: they are essentially equivalent in the Euclidean setup without constraints. The method presented here enjoys a number of variants (see, e.g., discussions in (Tseng, 2008)), some of which may involve two projections per iteration, as in (Nesterov, 2005, Section 3), (Lan *et al.*, 2011, Section 3). The method can also naturally be embedded with a backtracking procedure, exactly as in previous sections. Finally, such methods can be adapted to strong convexity, either on f, as in (Gasnikov and Nesterov, 2018) or on h, as in (Diakonikolas and Guzmán, 2021).

Remark 4.11. Note that the technique for rendering accelerated methods monotone (see Section 4.7.3) also directly applies to Algorithm 22.

Remark 4.12. Beyond the Euclidean setting where $w(x) = \frac{1}{2}\|.\|_2^2$, a classical example of the impact of the non-Euclidean setup optimizes a simple function over the simplex. In this case, writing $x^{(i)}$ for the ith component of some $x \in \mathbb{R}^d$, we consider the situation where h is the indicator function of the simplex

$$h(x) = \begin{cases} 0 & \text{if } \sum_{i=1}^d x^{(i)} = 1,\ x^{(i)} \geq 0\,(i = 1, \dots, d) \\ +\infty & \text{otherwise,} \end{cases}$$

and w is the *entropy* (which is closed, convex, and proper, and 1-strongly convex over the simplex for $\|.\| = \|.\|_1$; this is known as Pinsker's inequality). That is, define some $w_i : \mathbb{R} \to \mathbb{R}$:

$$w_i(x) = \begin{cases} x \log x & \text{if } x > 0, \\ 0 & \text{if } x = 0, \\ +\infty & \text{otherwise,} \end{cases}$$

and set $w(x) = \sum_{i=1}^d w_i(x^{(i)})$. In this case, the expression for the Bregman proximal gradient step in Algorithm 22 can be computed exactly, assuming $y_k^{(i)} \neq 0$ (for all $i = 1, ..., d$):

$$z_{k+1}^{(i)} = \frac{y_k^{(i)} \exp\left[-\frac{a_k}{L}[\nabla f(y_k)]^{(i)}\right]}{\sum_{i=1}^d y_k^{(i)} \exp\left[-\frac{a_k}{L}[\nabla f(y_k)]^{(i)}\right]}.$$

Hence, we also have that $y_k^{(i)} \neq 0$ as long as $y_0^{(i)} \neq 0$; a common technique is to instantiate $x_0^{(i)} = \frac{1}{d}$.

In this setup, a non-Euclidean geometry often provides a significant practical advantage when optimizing large-scale functions by improving the dependence from d to $\ln d$ in the final complexity bound. In fact, here we have $D_w(x_\star, x_0) \leq \ln d$ compared to

$D_{\frac{1}{2}\|.\|_2^2}(x_\star, x_0) \leq \frac{1}{2}$ in the Euclidean case, so the dependence in d is seemingly better in the Euclidean case. However, the choice of norms has a very significant impact. When the gradient has a small Lipschitz constant measured with respect to $\|.\| = \|.\|_1$, that is,

$$\|\nabla f(x) - \nabla f(y)\|_\infty \leq L_1 \|x - y\|_1,$$

the Lipschitz constant might be up to d times smaller than the constant computed using the Euclidean norm $\|.\| = \|.\|_2$ (using norm equivalences), i.e.,

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L_2 \|x - y\|_2,$$

with $L_1 \sim L_2/d$. The final complexity bound using a Euclidean geometry then reads

$$F(x_N) - F_\star \leq \frac{2L_1 d}{N^2},$$

as compared to

$$F(x_N) - F_\star \leq \frac{4L_1 \ln d}{N^2}$$

using the geometry induced by the entropy. The impact of the choice of norm is discussed extensively in, e.g. (Juditsky *et al.*, 2009, Example 2.1) and (d'Aspremont *et al.*, 2018).

Another related example is optimizing over a spectrahedron; see, for example, the nice introduction by Bubeck (2015, Chapter 4). This setup is largely motivated in (Nesterov, 2005). We refer to (Allen-Zhu and Orecchia, 2017; Diakonikolas and Guzmán, 2021) and the references therein for further discussions on this topic.

4.8 Continuous-time Interpretations

Before concluding this section, we survey another last popular approach to Nesterov's acceleration. The idea is to study continuous versions of first-order schemes, which enables simpler (less technical) proofs to emerge. One notorious caveat of such approaches is that they usually defer implementation details (i.e., discretization) to integration solvers (such as explicit Euler's scheme), but the simplicity of the proofs arguably renders this approach worth investigating. In particular, we see later that one usually does not use *smoothness* when computing worst-case convergence speeds. That is, smoothness is intrinsically linked to the discretization procedure, and not at all to the convergence speed of the continuous-time processes.

The section is divided in three parts. We start by reviewing results related to the *gradient flow*, a natural ordinary differential equation (ODE) for modelling first-order methods. Then, we continue with Nesterov's ODE, the continuous-time limit of Nesterov's accelerated gradient method as the step size vanishes (Su *et al.*, 2014). We conclude with discussions and pointers to different results and research directions relying on ODE interpretations of Nesterov's method.

For simplicity purposes, we make the choice of only presenting *informal* arguments for obtaining the continuous-time versions of gradient descent and of Nesterov's acceleration.

4.8.1 Gradient Flow

A starting point for linking first-order methods to continuous-time processes is to minimize a convex function f by following its *gradient flow*

$$\begin{cases} \dot{x}(t) = -\nabla f(x(t)), \\ x(0) = x_0 \end{cases} \tag{4.27}$$

where $\dot{x}(t) = \frac{dx(t)}{dt}$ denotes the usual time derivative of x, and where the time $t \in \mathbb{R}$ takes the role of the usual iteration counter $k \in \mathbb{N}$. For technical reasons, we also assume throughout that f is L-smooth, as it ensures the existence of a solution to (4.27).

From (4.27), one can recover usual first-order methods by playing with different integration schemes. Hence, first-order methods can typically be interpreted in the light of integration schemes, through specific notions of numerical stability, see, e.g., (Scieur *et al.*, 2017b). In short, whereas integration schemes typically aim at tracking the full trajectory of an ODE, optimization methods only target tracking the stationary point of (4.27) thereby requiring less stringent notions of numerical stability.

As a particular case, the classical explicit Euler integration scheme applied to (4.27) boils down to gradient descent for minimizing $f(x)$. On the other hand, (4.27) can be obtained as a natural continuous-time counterpart to gradient descent with vanishing step size. That is, gradient descent with step size Δ can be written as

$$\frac{x_{k+1} - x_k}{\Delta} = -\nabla f(x_k).$$

Informally, assume that the sequence $\{x_k\}_k$ is obtained as an approximation to a solution $x(t)$ to an ODE, with $x(0) = x_0$ and $x(t) \approx x(k\Delta)$. It is clear that

$$x_{k+1} \approx x(t + \Delta) = x(t) + \Delta \dot{x}(t) + o(\Delta),$$

and hence

$$-\nabla f(x(t)) = \frac{x(t + \Delta) - x(\Delta)}{\Delta} = \dot{x}(t) + \tfrac{o(\Delta)}{\Delta},$$

leading to (4.27) by taking the limit $\Delta \to 0$ on both sides.

Before moving to Nesterov's ODE, we glance at the convergence speed of the gradient flow towards an optimum when f is a convex function. For doing that, we use a similar potential function as that used in the discrete setup.

Convergence speed. We proceed essentially as in the previous section, but we see below that the proofs are much less technical. We introduce the (continuous) potential function:

$$\phi(t) \triangleq a(t) \left(f(x(t)) - f(x_\star) \right) + \|x(t) - x_\star\|_2^2.$$

The analysis simply consists in showing that $\dot{\phi}(t) \leq 0$. Indeed, just as in the discrete setup, we can then write:

$$\phi(t) \leq \phi(0),$$

with $\phi(t) \geq a(t)\left(f(x(t)) - f(x_\star)\right)$ and $\phi(0) = \|x(t) - x_\star\|_2^2$. Thereby, we reach

$$f(x(t)) - f(x_\star) \leq \frac{\phi(0)}{a(t)} = \frac{\|x(t) - x_\star\|_2^2}{a(t)},$$

and the worst-case speed of convergence of $f(x(t))$ towards $f(x_\star)$ is dictated by the growth rate of $a(t)$.

Theorem 4.27. Let f be a closed proper convex function, $x_\star \in \operatorname{argmin}_x f(x)$, and let $a(t) = t$. For any $x(\cdot)$ solution to (4.27) and any $t \geq 0$, it holds that

$$\dot{\phi}(t) \leq 0,$$

with

$$\phi(t) \triangleq a(t)\left(f(x(t)) - f(x_\star)\right) + \|x(t) - x_\star\|_2^2.$$

Proof. The proof simply uses a convexity inequality between $x(t)$ and $x_\star$. That is, explicit computations allow obtaining:

$$\dot{\phi}(t) = \dot{a}(t)\left(f(x(t)) - f(x_\star)\right) - a(t)\|\nabla f(x(t))\|_2^2 - \langle \nabla f(x(t)); x(t) - x_\star \rangle.$$

Then, a convexity inequality

$$f(x(t)) - f(x_\star) \leq \langle \nabla f(x(t)); x(t) - x_\star \rangle$$

allows writing

$$\dot{\phi}(t) \leq -a(t)\|\nabla f(x(t))\|_2^2 + (\dot{a}(t) - 1)\langle \nabla f(x(t)); x(t) - x_\star \rangle,$$

and we reach the desired inequality using $\dot{a}(t) = 1$:

$$\dot{\phi}(t) \leq -a(t)\|\nabla f(x(t))\|_2^2 \leq 0. \quad \blacksquare$$

Corollary 4.28. Let f be a closed proper convex function and $x_\star \in \operatorname{argmin}_x f(x)$. For any $x(\cdot)$ solution to (4.27) and any $t \geq 0$, it holds that

$$f(x(t)) - f(x_\star) \leq \frac{\|x(0) - x_\star\|_2^2}{t}.$$

Proof. The proof follows from $\dot{\phi}(t) \leq 0$ (Theorem 4.27) and

$$f(x(t)) - f(x_\star) \leq \frac{\phi(0)}{a(t)} = \frac{\|x(0) - x_\star\|_2^2}{a(t)},$$

with $a(t) = t$. $\blacksquare$

4.8.2 An Ordinary Differential Equation for Nesterov's Method

In this section, we briefly study a different ODE, namely

$$\begin{cases} \ddot{y}(t) + \frac{3}{t+T}\dot{y}(t) + \nabla f(y(t)) = 0, \\ y(0) = x_0, \\ \dot{y}(0) = 0, \end{cases} \tag{4.28}$$

where $T \geq 0$ is some constant. This ODE with $T = 0$ was proposed in (Su *et al.*, 2014) as the continuous-time interpretation of Nesterov's method. The case $T > 0$ might be considered instead for simplicity of the exposition, rendering the ODE directly well-defined even for $t = 0$. For simplicity, we also consider Algorithm 12 with $a_k = \frac{k}{2}$ for $k = 0, 1, \ldots$ (leading to $A_k = \frac{k^2}{4}$), that is,

$$\begin{aligned} x_{k+1} &= y_k - \Delta \nabla f(y_k) \\ y_{k+1} &= x_{k+1} + \frac{k}{k+3}(x_{k+1} - x_k), \end{aligned} \tag{4.29}$$

for $k = 0, 1, \ldots$. In this setup, the method can also be described in terms of a single sequence:

$$\begin{aligned} y_{k+1} = y_k - \Delta \nabla f(y_k) + \frac{k}{k+3}(y_k - y_{k-1}) \\ - \frac{k}{k+3}\Delta(\nabla f(y_k) - \nabla f(y_{k-1})). \end{aligned} \tag{4.30}$$

As shown in (Su *et al.*, 2014), it turns out that (4.28) can be seen as the continuous-time limit of (4.29) as $\Delta \to 0$. Using a similar informal development to that of (Su *et al.*, 2014), a few approximations allow to arrive to the desired ODE. For doing that, let us denote by $y(t)$ a trajectory of the limiting ODE, and let us assume that it is approximated by Nesterov's method (acting as a numerical integrator) as $y(t) \approx y_{t/\sqrt{\Delta}}$ (i.e., we make the correspondence $y\left(k\sqrt{\Delta}\right) \approx y_k$). Note the time scaling in $\sqrt{\Delta}$ instead of Δ due to the second-order dynamics of the system. Next, we use the following Taylor expansions

$$\begin{aligned} \frac{y\left(t+\sqrt{\Delta}\right) - y(t)}{\sqrt{\Delta}} &= \dot{y}(t) + \frac{\sqrt{\Delta}}{2}\ddot{y}(t) + o\left(\sqrt{\Delta}\right) \\ \frac{y\left(t-\sqrt{\Delta}\right) - y(t)}{\sqrt{\Delta}} &= -\dot{y}(t) + \frac{\sqrt{\Delta}}{2}\ddot{y}(t) + o\left(\sqrt{\Delta}\right) \\ \sqrt{\Delta}\nabla f\left(y\left(t+\sqrt{\Delta}\right)\right) &= \sqrt{\Delta}\nabla f(y(t)) + o\left(\sqrt{\Delta}\right) \\ \frac{t}{t+3\sqrt{\Delta}} &= 1 - 3\frac{\sqrt{\Delta}}{t} + o\left(\sqrt{\Delta}\right) \end{aligned}$$

for writing

$$\begin{aligned} &\dot{y}(t) + \frac{\sqrt{\Delta}}{2}\ddot{y}(t) + o\left(\sqrt{\Delta}\right) \\ &\quad = \left(1 - 3\frac{\sqrt{\Delta}}{t}\right)\left(\dot{y}(t) - \frac{\sqrt{\Delta}}{2}\ddot{y}(t) + o\left(\sqrt{\Delta}\right)\right) - \sqrt{\Delta}\nabla f(y(t)). \end{aligned}$$

Simplifying these expressions, dividing all terms by $\sqrt{\Delta}$ and taking the limit $\Delta \to 0$ leads to the desired

$$\ddot{y}(t) + \frac{3}{t}\dot{y}(t) + \nabla f(y(t)) = 0.$$

Convergence speed. As for the gradient flow, the analysis only requires an appropriate (continuous) potential function $\phi(t)$. The desired conclusion follows from showing that $\dot{\phi}(t) < 0$, as provided by the following theorem.

Theorem 4.29. Let f be a closed proper convex function and $x_\star \in \operatorname{argmin}_x f(x)$. For any $x(\cdot)$ solution to (4.28) and any $t \geq 0$, it holds that

$$\dot{\phi}(t) \leq 0,$$

with

$$\phi(t) \triangleq (t+T)^2(f(x(t)) - f_\star) + 2\|x(t) + \tfrac{t+T}{2}\dot{x}(t) - x_\star\|_2^2.$$

Proof. From the expression of $\phi(t)$, it is relatively straightforward to obtain that

$$\begin{aligned}\dot{\phi}(t) =& 2(t+T)(f(x(t)) - f(x_\star)) + (t+T)^2\langle \nabla f(x(t)); \dot{x}(t)\rangle \\ &+ 4\langle (x(t) + \frac{t+T}{2}\dot{x}(t) - x_\star); \frac{3}{2}\dot{x}(t) + \frac{t+T}{2}\ddot{x}(t)\rangle.\end{aligned}$$

Using (4.28), one can substitute the expression of $\ddot{x}(t)$, leading to

$$\dot{\phi}(t) = 2(t+T)(f(x(t)) - f(x_\star)) + 2(t+T)\langle \nabla f(x(t)); x_\star - x(t)\rangle$$

and it follows from convexity that $\dot{\phi}(t) \leq 0$, as desired. ■

Corollary 4.30. Let f be a closed proper convex function and $x_\star \in \operatorname{argmin}_x f(x)$. For any $x(\cdot)$ solution to (4.28) and any $t \geq 0$, it holds that

$$f(x(t)) - f(x_\star) \leq \frac{T^2(f(x(0)) - f(x_\star)) + \|x(0) - x_\star\|_2^2}{(t+T)^2}.$$

Proof. The proof follows from $\dot{\phi}(t) \leq 0$ (Theorem 4.29), implying $\phi(t) \leq \phi(0)$ and thereby

$$f(x(t)) - f(x_\star) \leq \frac{\phi(0)}{(t+T)^2}.$$ ■

4.8.3 Continuous-time Approaches to Acceleration: Summary

In this section, we saw that some ordinary differential equations can be used for modelling gradient and accelerated gradient-type methods. The corresponding convergence proofs are much simpler, as they only involve using a single inequality, namely convexity between two points: $x(t)$ and $x_\star$. However, convergence speeds of continuous-time versions of algorithms might not be representative of their behaviors, as the corresponding ODE might be complicated to integrate, and as using numerical integration solvers might break the potentially nice convergence properties of the continuous-time dynamics.

The content of this section is explored at length in many references, see e.g., (Su *et al.*, 2014; Krichene *et al.*, 2015; Wibisono *et al.*, 2016; Attouch *et al.*, 2018). We discuss a few extensions and limitations below, before concluding the section.

Integration methods and optimization. Continuous-time formulations of gradient-based methods cannot be implemented as is on digital computers. Therefore, continuous-time analyses cannot provide a complete picture on the topic without incorporating numerical integration methods into the analyses. Another symptom of this incompleteness is that of different optimization methods giving rise to the same limiting ODEs. For instance, the same limiting ODE is obtained from Polyak's heavy-ball, from Nesterov's accelerated methods, and from the triple momentum methods; see (Shi *et al.*, 2021) and (Sun *et al.*, 2020).

This observation motivates different lines of works. In (Scieur *et al.*, 2017b), the authors focus on the gradient flow (4.27) and propose different integration methods for recovering classical first-order methods in the quadratic minimization setup. In (Su *et al.*, 2014), it is shown that forward Euler integration of Nesterov's ODE (4.28) leads to a heavy-ball type method, close to Nesterov's acceleration. In (Shi *et al.*, 2019), the authors obtain accelerated methods by integrating "high resolution" variants of the *accelerated ODEs* via symplectic methods (partially implicit, and partially explicit integration rules). Various discussions, developments, and connections between continuous-time systems and their discrete counterparts are further presented in Diakonikolas and Orecchia (2019b), Siegel (2019), and Sanz Serna and Zygalakis (2021).

Strongly convex ODEs. In the strongly convex case, limiting ODEs for "stationnary" accelerated methods such as Nesterov's method with constant momentum (Algorithm 15) or triple momentum method (Algorithm 18) are also presented in different works; see, e.g., (Shi *et al.*, 2021; Sun *et al.*, 2020).

Continuized methods. As previously discussed, it might not be simple to use classical continuous-time methods on digital computers (nontrivial integration schemes must be deployed). A family of so-called "*continuized methods*" are directly implementable while keeping the benefits of the continuous-time approaches. Those methods rely on *randomized* discretizations of the continuous-time process (Even *et al.*, 2021).

4.9 Notes and References

Estimate sequences, potential functions, and differential equations. Potential functions were already used in the original paper by Nesterov (1983) to develop accelerated methods. Nesterov (2003) developed estimate sequences as an alternate, more constructive, approach to obtaining optimal first-order methods. Since then, both approaches have been used in many references on this topic, in a variety of settings. Tseng (2008) provides a helpful unified view of accelerated methods. Estimate sequences have been extensively studied by, e.g., Nesterov (2013), Baes (2009), Devolder (2011), and Kulunchakov and Mairal (2020). Another related approach is that of the *approximate duality gap* Diakonikolas and Orecchia (2019b) which is a constructive approach to estimate sequences/potential functions with a continuous-time counterpart.

Beyond Euclidean geometries. *Mirror descent* dates back to the work of Nemirovsky and Yudin (1983a). It has been further developed and used in many subsequent works (Ben-Tal and Nemirovsky, 2001; Nesterov, 2005; Nesterov, 2009; Xiao, 2010; Juditsky and Nesterov, 2014; Diakonikolas and Guzmán, 2021). Sound pedagogical surveys can be found in (Beck and Teboulle, 2003; Juditsky and Nemirovsky, 2011a; Juditsky and Nemirovsky, 2011b; Bubeck, 2015).

Beyond the setting described in this section, mirror descent has also been studied in the *relative smoothness* setting, introduced by Bauschke *et al.* (2016)—see also (Teboulle, 2018)—, and extended to the notion of *relative strong convexity* by Lu *et al.* (2018). However, acceleration remains an open issue in the context of relative smoothness and relative strong convexity, and it is generally unclear which additional assumptions allow accelerated rates. It is, however, clear that additional assumptions are required, as emphasized by the lower bound provided by Dragomir *et al.* (2021). In particular, accelerated schemes are known under an additional *triangle scaling inequality* (Hanzely *et al.*, 2021; Gutman and Peña, 2018).

Lower complexity bounds. Lower complexity bounds have been studied in a variety of settings to establish limits on the worst-case performance of black-box methods. The classical reference on this topic is the book by Nemirovsky and Yudin (1983c).

Of particular interest to us, Nemirovsky (1991) and Nemirovsky (1992) establish the optimality of the Chebyshev and of the conjugate gradient methods for convex quadratic minimization; see, also, a complete picture provided in the course notes by Nemirovsky (1994). Lower bounds for black-box first-order methods in the context of smooth convex and smooth strongly convex optimization can be found in (Nesterov, 2003). The final lower bound for black-box smooth convex minimization was obtained by Drori (2017); it demonstrates the optimality of the optimized gradient method, as well as that of conjugate gradients, as discussed earlier in this section. Lower bounds for ℓ_p norms in the mirror descent setup are constructed in Guzmán and Nemirovsky (2015), whereas a lower bound for mirror descent in the relative smoothness setup is provided by Dragomir *et al.* (2021).

Changing the performance measure. Obtaining (practical) accelerated method for other types of convergence criteria, such as gradient norms, is still not a fully settled issue. These criterion are important in other contexts, including dual methods, and can be used to draw links between methods intrinsically designed to solve convex problems and those used in nonconvex settings, where the goal is to find stationary points. There are a few *tricks* that make it possible to pass from a guarantee in one context to another one. For example, a regularization trick was proposed in (Nesterov, 2012b) that yields approximate solutions with small gradient norm. Beyond that, in the context of smooth convex minimization, recent progresses have been made by Kim and Fessler (2020), who designed an optimized method for minimizing the gradient norm after a given number

of iterations. This method was analyzed through potential functions in (Diakonikolas and Wang, 2021), and its geometric structure was further explored and exploited in (Lee *et al.*, 2021). Corresponding lower bounds, based on quadratic minimization, for a variety of performance measures can be found in (Nemirovsky, 1992).

Adaptation and backtracking line-searches. The idea of using backtracking line-searches is classical and is attributed to Goldstein (1962) and Armijo (1966); see also discussions in (Nocedal and Wright, 2006; Bonnans *et al.*, 2006). It was already incorporated in the original work of Nesterov (1983) to estimate the smoothness constant within an accelerated gradient method. Since then, many works on the topic have relied heavily on this technique, which is often adapted to obtain better practical performance; see, for example, (Scheinberg *et al.*, 2014; Chambolle and Pock, 2016; Florea and Vorobyov, 2018; Calatroni and Chambolle, 2019). A more recent adaptive step size strategy (without line-search) can be found in (Malitsky and Mishchenko, 2020).

Numerical stability, inexactness, stochasticity, and randomness. The ability to use approximate first-order information, be it stochastic or deterministic, is key for tackling certain problems for which computing the exact gradient is expensive. Deterministic (or adversarial) error models are studied in, e.g., (d'Aspremont, 2008; Schmidt *et al.*, 2011; Devolder *et al.*, 2014; Devolder, 2013; Devolder *et al.*, 2013; Aybat *et al.*, 2020) through different noise models. Such approaches can also be deployed when the projection/proximal operation is computed approximately (Schmidt *et al.*, 2011; Villa *et al.*, 2013) (see also Section 5 and the references therein).

Similarly, stochastic approximations and incremental gradient methods are key in many statistical learning problems, where samples are accessed one at a time and for which it is not desirable to optimize beyond the data accuracy (Bottou and Bousquet, 2007). For this reason, the old idea of stochastic approximations (Robbins and Monro, 1951) is still widely used and remains an active area of research. The "optimal" variants of stochastic approximations were developed much later (Lan, 2008) with the rise of machine learning applications. In this context, it is not possible to asymptotically accelerate convergence rates but only to accelerate the transient phase toward a purely stochastic regime; see also (Hu *et al.*, 2009; Xiao, 2010; Devolder, 2011; Lan, 2012; Dvurechensky and Gasnikov, 2016; Aybat *et al.*, 2019; Gorbunov *et al.*, 2020)—in particular, we note that "stochastic" estimate sequences were developed in (Devolder, 2011; Kulunchakov and Mairal, 2020). The case of stochastic noise arising from sampling an objective function that is a *finite sum* of smooth components attracted substantial attention in the 2010s, starting with (Schmidt *et al.*, 2017; Johnson and Zhang, 2013; Shalev-Shwartz and Zhang, 2013; Defazio *et al.*, 2014a; Defazio *et al.*, 2014b; Mairal, 2015) and was then extended to feature acceleration techniques (Shalev-Shwartz and Zhang, 2014; Allen-Zhu, 2017; Zhou *et al.*, 2018; Zhou *et al.*, 2019). Acceleration techniques also apply in the context of randomized block coordinate descent; see, for example, Nesterov (2012a), Lee and Sidford (2013), Fercoq and Richtárik (2015), and Nesterov and Stich (2017).

Higher-order methods. Acceleration mechanisms have also been proposed in the context of higher-order methods. This line of work started with the cubic regularized Newton method introduced in (Nesterov and Polyak, 2006) and its acceleration using estimate sequence mechanisms (Nesterov, 2008); see also (Baes, 2009; Wilson *et al.*, 2021) and (Monteiro and Svaiter, 2013) (which we also discuss in the next section). Optimal higher-order methods were presented by (Gasnikov *et al.*, 2019). It was not clear before the work of Nesterov (2019) that intermediate subproblems arising in the context of higher-order methods were tractable. The fact that tractability is not an issue has attracted significant attention to these methods.

Optimized methods. Optimized gradient methods were discovered by Kim and Fessler (2016), based on the work by Drori and Teboulle (2014). Since then, optimized methods have been studied in various settings: incorporating constraints/proximal terms (Kim and Fessler, 2018b; Taylor *et al.*, 2017a); optimizing gradient norms (Kim and Fessler, 2018c; Kim and Fessler, 2020; Diakonikolas and Wang, 2021) (as an alternative to (Nesterov, 2012b)); adapting to unknown problem parameters using exact line-searches (Drori and Taylor, 2020) or restarts (Kim and Fessler, 2018a); and in the strongly convex case (Van Scoy *et al.*, 2017; Cyrus *et al.*, 2018; Park *et al.*, 2021; Taylor and Drori, 2021). Such methods have also appeared in the context of fixed-point iterations (Lieder, 2021) and proximal methods (Kim, 2021; Barré *et al.*, 2020a).

On obtaining proofs from this section. The worst-case performance of first-order methods can often be computed numerically. This has been shown in (Drori and Teboulle, 2014; Drori, 2014; Drori and Teboulle, 2016; Taylor *et al.*, 2017c) through the introduction of performance estimation problems. Such techniques might be framed in different ways, e.g., from a purely optimization-based point of view (Drori and Teboulle, 2014; Taylor *et al.*, 2017c) or from a control-theoretical perspective (Lessard *et al.*, 2016; Fazlyab *et al.*, 2018). We provide a brief summary in the following lines with more details in Appendix C.

The performance estimation approach was shown to provide *tight* certificates, from which one can recover both worst-case certificates and matching worst-case problem instances in (Taylor *et al.*, 2017c; Taylor *et al.*, 2017a). One consequence is that worst-case guarantees for first-order methods such as those detailed in this section can *always* be obtained as a weighted sum of the appropriate inequalities characterizing the problem at hand; see, for instance, (De Klerk *et al.*, 2017; Dragomir *et al.*, 2021). A similar approach framed in control theoretic terms, and originally tailored to obtain geometric convergence rates, was developed by Lessard *et al.* (2016) and can also be used to form potential functions (Hu and Lessard, 2017) as well as optimized methods such as the triple momentum method (Van Scoy *et al.*, 2017; Cyrus *et al.*, 2018; Lessard and Seiler, 2020).

The proofs in this section were obtained by using the performance estimation approach tailored for potential functions (Taylor and Bach, 2019) together with the performance

estimation toolbox (Taylor *et al.*, 2017b). In particular, the potential function for the optimized gradient method can be found in (Taylor and Bach, 2019, Theorem 11) (see also (Taylor and Drori, 2021; Park *et al.*, 2021)). These techniques can be used for to either validate or rediscover the proofs in this section numerically, through semidefinite programming. More details are provided in Appendix C.

For the purpose of reproducibility, we provide the corresponding code, as well as notebooks for numerically and symbolically verifying the algebraic reformulations in this section at https://github.com/AdrienTaylor/AccelerationMonograph.

5

Proximal Acceleration and Catalysts

In this section, we present simple methods based on approximate proximal operations that produce accelerated gradient-based methods. This idea is exploited for example in the Catalyst (Lin *et al.*, 2015; Lin *et al.*, 2018) and Accelerated Hybrid Proximal Extragradient (A-HPE) (Monteiro and Svaiter, 2013) frameworks. In essence, the idea is to develop (conceptual) accelerated proximal point algorithms and to use classical iterative methods to approximate the proximal point. In particular, these frameworks produce accelerated gradient methods (in the same sense as Nesterov's acceleration) when the approximate proximal points are computed using linearly converging gradient-based optimization methods.

5.1 Introduction

We review acceleration from the perspective of proximal point algorithms (PPA). The key concept here, called *proximal operation*, dates back to the 1960s, with the works of Moreau (1962; 1965). Its introduction to optimization is attributed to Martinet (1970; 1972) and was primarily motivated by its link with augmented Lagrangian techniques. In contrast with previous sections, where information about the functions to be minimized was obtained through their gradients, the following sections deal with the case in which information is gathered through a *proximal operator* or an approximation of that operator.

The proximal point algorithm and its use in the development of optimization schemes are surveyed in (Parikh and Boyd, 2014). We aim to go in a slightly different direction here and describe the use of the PPA in an outer loop to obtain improved convergence guarantees in the spirit of the Accelerated Hybrid Proximal Extragradient (A-HPE) method (Monteiro and Svaiter, 2013) and of Catalyst (Lin *et al.*, 2015; Lin *et al.*, 2018).

In this section, we focus on the problem of solving

$$f_\star = \min_{x \in \mathbb{R}^d} f(x), \tag{5.1}$$

where f is closed, convex, and proper (it has a closed convex non-empty epigraph), which we denote by $f \in \mathcal{F}_{0,\infty}$ in line with Definition 4.1 from Section 4. We denote by $\partial f(x)$ the subdifferential of f at $x \in \mathbb{R}^d$ and by $g_f(x) \in \partial f(x)$ some element of the subdifferential at x, irrespective of whether f is continuously differentiable. We aim to find an ϵ-approximate solution x such that $f(x) - f_\star \leq \epsilon$.

It is possible to develop optimized proximal methods in the spirit of the optimized gradient methods presented in Section 4. That is, given a computational budget—in the proximal setting, this consists of a number of iterations and a sequence of step sizes—one can choose the algorithmic parameters to optimize the worst-case performance. The proximal equivalent to the optimized gradient method is Güler's second method (Güler, 1992, Section 6) (see the discussions in Section 5.6). We do not spend time on this method here and directly aim for methods designed from simple potential functions, in the same spirit our approach to Nesterov's accelerated gradient methods in Section 4.

5.2 Proximal Point Algorithm and Acceleration

Whereas the base method for minimizing a function using its gradient is gradient descent:

$$x_{k+1} = x_k - \lambda g_f(x_k),$$

the base method for minimizing a function using its proximal oracle is the proximal point algorithm:

$$x_{k+1} = \operatorname{prox}_{\lambda f}(x_k), \tag{5.2}$$

where the proximal operator is given by

$$\operatorname{prox}_{\lambda f}(x) \triangleq \operatorname*{argmin}_y \{\Phi(y; x) \triangleq \lambda f(y) + \frac{1}{2}\|y - x\|_2^2\}.$$

The proximal point algorithm has a number of intuitive interpretations, with two of them being particularly convenient for our purposes.

- Optimality conditions of the proximal subproblem reveal that a proximal step corresponds to an implicit (sub)gradient method:

  $$x_{k+1} = x_k - \lambda g_f(x_{k+1}),$$

  where $g_f(x_{k+1}) \in \partial f(x_{k+1})$.

- Using the proximal point algorithm is equivalent to applying gradient descent to the Moreau envelope of f, where the Moreau envelope, denoted F_λ, is provided by

  $$F_\lambda(x) \triangleq \min_y \{f(y) + \frac{1}{2\lambda}\|y - x\|_2^2\}.$$

The Moreau envelope has the same set of optimal solutions as f, while enjoying attractive additional regularity properties (it is $1/\lambda$-smooth and convex; see Definition 4.1). More precisely, its gradient is given by

$$\nabla F_\lambda(x) = \left(x - \operatorname{prox}_{\lambda f}(x)\right)/\lambda,$$

(see (Lemaréchal and Sagastizábal, 1997) for more details). This allows us to write

$$x_{k+1} = \operatorname{prox}_{\lambda f}(x_k) = x_k - \lambda \nabla F_\lambda(x_k),$$

and hence to write proximal minimization methods (as well as their inexact and accelerated variants) applied to f as classical gradient methods (and their inexact and accelerated variants) applied to F_λ.

In general, proximal operations are expensive, sometimes nearly as expensive as minimizing the function itself. However, there are many cases, especially in the context of composite optimization problems, where one can isolate parts of the objective for which proximal operators actually have analytical solutions; see, e.g. (Chierchia *et al.*, 2020) for a list of such examples.

In the following sections, we start by analyzing such proximal point methods, and then at the end of the section we show how proximal methods can be used in *outer loops*, where proximal subproblems are solved approximately using a classical iterative method (in inner loops). In particular, we describe how this combination produces accelerated numerical schemes.

5.2.1 Convergence Analysis

Given the links between proximal operations and gradient methods, it is probably not surprising that proximal point methods for convex optimization can be analyzed using potential functions similar to those used for gradient methods.

However, there is a huge difference between gradient and proximal steps, as the latter can be made arbitrarily "powerful" by taking large step sizes. In other words, a single proximal operation can produce an arbitrarily good approximate solution by picking an arbitrarily large step size. This contrasts with gradient descent, where large step sizes make the method diverge. This fact is clarified later by Corollary 5.2. However, this nice property of proximal operators comes at a cost: we may not be able to efficiently compute the proximal step.

As emphasized by the next theorem, proximal point methods for solving (5.1) can be analyzed by using similar potentials as those of gradient-based methods. We use

$$\phi_k \triangleq A_k(f(x_k) - f(x_\star)) + \frac{1}{2}\|x_k - x_\star\|_2^2 \tag{5.3}$$

and show that $\phi_{k+1} \leq \phi_k$. As before, this type of reasoning can be used recursively:

$$\begin{aligned} A_N(f(x_N) - f(x_\star)) \leq \phi_N \leq \phi_{N-1} \leq \ldots \leq \phi_0 = & A_0(f(x_0) - f(x_\star)) \\ & + \frac{1}{2}\|x_0 - x_\star\|_2^2, \end{aligned} \tag{5.4}$$

thereby reaching bounds of the type $f(x_N) - f_\star \leq \frac{1}{2A_N}\|x_0 - x_\star\|_2^2 = O(A_N^{-1})$, assuming $A_0 = 0$. Since the convergence rates are dictated by the growth rate of the scalar sequence $\{A_k\}_k$, the proofs are designed to increase A_k as fast as possible.

Theorem 5.1. Let $f \in \mathcal{F}_{0,\infty}$. For any $k \in \mathbb{N}$, $A_k, \lambda_k \geq 0$ and any x_k, it holds that

$$\begin{aligned} A_{k+1}(f(x_{k+1}) - f(x_\star)) + \frac{1}{2}\|x_{k+1} - x_\star\|_2^2 \\ \leq A_k(f(x_k) - f(x_\star)) + \frac{1}{2}\|x_k - x_\star\|_2^2, \end{aligned}$$

with $x_{k+1} = \operatorname{prox}_{\lambda_k f}(x_k)$ and $A_{k+1} = A_k + \lambda_k$.

Proof. We perform a weighted sum of the following valid inequalities originating from our assumptions.

- Convexity between x_{k+1} and $x_\star$ with weight λ_k:
$$f(x_\star) \geq f(x_{k+1}) + \langle g_f(x_{k+1}); x_\star - x_{k+1}\rangle,$$
with some $g_f(x_{k+1}) \in \partial f(x_{k+1})$.
- Convexity between x_{k+1} and x_k with weight A_k:
$$f(x_k) \geq f(x_{k+1}) + \langle g_f(x_{k+1}); x_k - x_{k+1}\rangle,$$
with the same $g_f(x_{k+1}) \in \partial f(x_{k+1})$ as before.

By performing a weighted sum of these two inequalities with their respective weights, we obtain the following valid inequality:

$$\begin{aligned} 0 \geq &\lambda_k \left[f(x_{k+1}) - f(x_\star) + \langle g_f(x_{k+1}); x_\star - x_{k+1}\rangle\right] \\ &+ A_k \left[f(x_{k+1}) - f(x_k) + \langle g_f(x_{k+1}); x_k - x_{k+1}\rangle\right]. \end{aligned}$$

By matching the expressions term by term and by substituting $x_{k+1} = x_k - \lambda_k g_f(x_{k+1})$, one can easily check that the previous inequality can be rewritten exactly as

$$\begin{aligned} (A_k + \lambda_k)(f(x_{k+1}) - f(x_\star)) + \frac{1}{2}\|x_{k+1} - x_\star\|_2^2 \\ \leq A_k(f(x_k) - f(x_\star)) + \frac{1}{2}\|x_k - x_\star\|_2^2 - \lambda_k \frac{2A_k + \lambda_k}{2}\|g_f(x_{k+1})\|_2^2. \end{aligned}$$

By omitting the last term on the right hand-side (which is nonpositive), we reach the desired statement. ∎

The first proof of the following worst-case guarantee is due to Güler (1991) and directly follows from the previous potential.

Corollary 5.2. Let $f \in \mathcal{F}_{0,\infty}$, $\{\lambda_i\}_{i\geq 0}$ be a sequence of nonnegative step sizes, and $\{x_i\}_{i\geq 0}$ be the sequence of iterates from the corresponding proximal point algorithm (5.2). For all $k \in \mathbb{N}$, $k \geq 1$, it holds that

$$f(x_k) - f_\star \leq \frac{\|x_0 - x_\star\|_2^2}{2\sum_{i=0}^{k-1} \lambda_i}.$$

Proof. It follows directly from the potential with the choice $A_0 = 0$. That is, we use the potential ϕ_k defined in (5.3) with Theorem 5.1, which allows using the chaining argument from (5.4). We obtain:

$$f(x_k) - f(x_\star) \leq \frac{1}{2A_k}\|x_0 - x_\star\|_2^2,$$

where $A_k = \sum_{i=0}^{k-1} \lambda_i$ and the claim directly follows. ∎

Note again that we can make this bound arbitrarily good by simply increasing the value of the λ_k. There is no contradiction here because the proximal oracle is massively stronger than the usual gradient step, as previously discussed. However, solving even a single proximal step is usually (nearly) as hard as solving the original optimization problem, so the proximal method, as detailed here, is a purely conceptual algorithm.

Note that the choice of a constant step size $\lambda_k = \lambda$ results in $f(x_N) - f_\star = O(N^{-1})$ convergence, reminiscent of gradient descent. It turns out that as for gradient-based optimization of smooth convex functions, it is possible to improve this result to $O(N^{-2})$ by using information from previous iterations. This idea was proposed by Güler (1992). In the case of a constant step size $\lambda_k = \lambda$, one possible way to obtain this improvement is to apply Nesterov's method (or any other accelerated variant) to the Moreau envelope of f. For varying step sizes, the corresponding bound has the form

$$f(x_k) - f_\star \leq \frac{2\|x_0 - x_\star\|_2^2}{\left(\sum_{i=0}^{k-1} \sqrt{\lambda_i}\right)^2}.$$

In addition, Güler's acceleration is actually robust to computation errors (as described in the next sections), while allowing for varying step size strategies.

These two key properties allow using Güler's acceleration to design improved numerical optimization schemes using

1. Approximate proximal steps, for example by approximately solving the proximal subproblems via iterative methods; and

2. The step size λ_i's can be increased from one iteration to the next, allowing for arbitrarily fast convergence rates to be achieved (assuming that the proximal subproblems can be solved efficiently).

It is important to note that the classical lower bounds for gradient-type methods do not apply here, as we use the much stronger, and more expensive, proximal oracle. It is therefore not a surprise that such techniques (i.e., increasing the step sizes λ_k from iteration to iteration) might beat the $O(N^{-2})$ bound obtained through Nesterov's acceleration. Such increasing step size rules can be used, for example, when solving the proximal subproblem via Newton's method, as proposed by Monteiro and Svaiter (2013).

5.3 Güler and Monteiro-Svaiter Acceleration

In this section, we describe an accelerated version of the proximal point algorithm which may involve inexact proximal evaluations. The method detailed below is a simplified version, sufficient for our purposes, of that of Monteiro and Svaiter (2013), and we provide a simple convergence proof for it. The method essentially boils down to that of Güler (1992) when exact proximal evaluations are used (note that the inexact analysis of Güler (1992) has a few gaps).

Before proceeding, we mention that there exist quite a few natural notions of inexactness for proximal operations. In this section, we focus on approximately satisfying the first-order optimality conditions of the proximal problem

$$x_{k+1} = \operatorname*{argmin}_x \{\Phi(x; y_k) \triangleq f(x) + \frac{1}{2\lambda_k}\|x - y_k\|_2^2\}.$$

In other words, optimality conditions of the proximal subproblem are

$$0 = \lambda_k g_f(x_{k+1}) + x_{k+1} - y_k,$$

for some $g_f(x_{k+1}) \in \partial f(x_{k+1})$. In the following lines, we instead tolerate an error e_k:

$$e_k = \lambda_k g_f(x_{k+1}) + x_{k+1} - y_k,$$

and require $\|e_k\|_2$ to be small enough to guarantee convergence—even starting at an optimal point does not imply staying at it, without proper assumptions on e_k. One possibility is to require $\|e_k\|_2$ to be small with respect to the distance between the starting point y_k and the approximate solution to the proximal subproblem x_{k+1}.

Formally, we use the following definition for an approximate solution with relative inaccuracy $0 \leq \delta \leq 1$:

$$x_{k+1} \approx_\delta \operatorname{prox}_{\lambda_k f}(y_k) \iff \left[\begin{array}{l} \|e_k\|_2 \leq \delta \|x_{k+1} - y_k\|_2 \\ \text{with } e_k \triangleq x_{k+1} - y_k + \lambda_k g_f(x_{k+1}) \\ \text{for some } g_f(x_{k+1}) \in \partial f(x_{k+1}) \end{array}\right] \tag{5.5}$$

Intuitively, this notion allows for tolerance of relatively large errors when the solution of the proximal subproblem is far from y_k (meaning that y_k is also far away from a minimum of f), while demanding relatively small errors when approaching a solution. On the other side, if y_k is an optimal point for f, then so is x_{k+1}, as shown by the following proposition.

Proposition 5.1. Let $y_k \in \operatorname{argmin}_x f(x)$. For any $\delta \in [0, 1]$ and any $x_{k+1} \approx_\delta \operatorname{prox}_{\lambda_k f}(y_k)$, it holds that $x_{k+1} \in \operatorname{argmin}_x f(x)$.

Proof. We only consider the case $\delta = 1$ since without loss of generality $x_{k+1} \approx_1 \operatorname{prox}_{\lambda_k f}(y_k) \Rightarrow x_{k+1} \approx_\delta \operatorname{prox}_{\lambda_k f}(y_k)$ for any $\delta \in [0, 1]$. By definition of x_{k+1}, we have

$$\begin{array}{rrcl} & \|x_{k+1} - y_k + \lambda_k g_f(x_{k+1})\|_2^2 & \leq & \|x_{k+1} - y_k\|_2^2 \\ \Leftrightarrow & 2\lambda_k \langle g_f(x_{k+1}); x_{k+1} - y_k \rangle & \leq & -\lambda_k^2 \|g_f(x_{k+1})\|_2^2, \end{array} \tag{5.6}$$

for some $g_f(x_{k+1}) \in \partial f(x_{k+1})$. (The second inequality follows from base algebraic manipulations of the first one.) In addition, optimality of y_k implies that

$$\langle g_f(x_{k+1});\, x_{k+1} - y_k\rangle = \langle g_f(x_{k+1}) - g_f(y_k);\, x_{k+1} - y_k\rangle \geq 0,$$

with $g_f(y_k) = 0 \in \partial f(y_k)$, where the second inequality follows from the convexity of f (see, e.g., Section A). Therefore, condition (5.6) can be satisfied only when $g_f(x_{k+1}) = 0$, meaning that x_{k+1} is a minimizer of f. ■

Now, assuming that it is possible to find an approximate solution to the proximal operator, one can use Algorithm 23, originally from (Monteiro and Svaiter, 2013), to minimize the convex function f. For simplicity, the parameters A_k, a_k in the algorithm are optimized for $\delta = 1$; they can be slightly improved by exploiting the case $0 \leq \delta < 1$.

Algorithm 23 An inexact accelerated proximal point method (Monteiro and Svaiter, 2013)

Input: A convex function f and an initial point x_0.

Initialize $z_0 = x_0$ and $A_0 = 0$.
for $k = 0, \ldots$ **do**
  Pick $a_k = \frac{\lambda_k + \sqrt{\lambda_k^2 + 4A_k\lambda_k}}{2}$
  $A_{k+1} = A_k + a_k$.
  $y_k = \frac{A_k}{A_k + a_k} x_k + \frac{a_k}{A_k + a_k} z_k$
  $x_{k+1} \approx_\delta \operatorname{prox}_{\lambda_k f}(y_k)$ (see Eq. (5.5), for some $\delta \in [0, 1]$)
  $z_{k+1} = z_k - a_k g_f(x_{k+1})$
end for

Output: Approximate solution x_{k+1}.

Perhaps surprisingly, this method can be analyzed with the same potential as the proximal point algorithm, despite the presence of computation errors.

Theorem 5.3. Let $f \in \mathcal{F}_{0,\infty}$. For any $k \in \mathbb{N}$, $A_k, \lambda_k \geq 0$ and any $x_k, z_k \in \mathbb{R}^d$, it holds that

$$A_{k+1}(f(x_{k+1}) - f(x_\star)) + \frac{1}{2}\|z_{k+1} - x_\star\|_2^2 \leq A_k(f(x_k) - f(x_\star)) + \frac{1}{2}\|z_k - x_\star\|_2^2,$$

where x_{k+1} and z_{k+1} are generated by one iteration of Algorithm 23, and $A_{k+1} = A_k + a_k$.

Proof. We perform a weighted sum of the following valid inequalities, which stem from our assumptions.

- Convexity between x_{k+1} and $x_\star$ with weight $A_{k+1} - A_k$:

$$f(x_\star) \geq f(x_{k+1}) + \langle g_f(x_{k+1}); x_\star - x_{k+1}\rangle,$$

  for some $g_f(x_{k+1}) \in \partial f(x_{k+1})$, which we also use below.

- Convexity between x_{k+1} and x_k with weight A_k:

$$f(x_k) \geq f(x_{k+1}) + \langle g_f(x_{k+1}); x_k - x_{k+1} \rangle.$$

- Error magnitude with weight $(A_{k+1})/(2\lambda_k)$:

$$\|e_k\|_2^2 \leq \|x_{k+1} - y_k\|_2^2,$$

  which is valid for all $\delta \in [0, 1]$ in (5.5).

By performing the weighted sum of these three inequalities, we obtain the following valid inequality:

$$\begin{aligned} 0 \geq & (A_{k+1} - A_k) \left[f(x_{k+1}) - f(x_\star) + \langle g_f(x_{k+1}); x_\star - x_{k+1} \rangle \right] \\ & + A_k \left[f(x_{k+1}) - f(x_k) + \langle g_f(x_{k+1}); x_k - x_{k+1} \rangle \right] \\ & + \frac{A_{k+1}}{2\lambda_k} \left[\|e_k\|_2^2 - \|x_{k+1} - y_k\|_2^2 \right]. \end{aligned}$$

After substituting $A_{k+1} = A_k + a_k$, $x_{k+1} = A_k/(A_k + a_k)x_k + a_k/(A_k + a_k)z_k - \lambda_k g_f(x_{k+1}) + e_k$ and $z_{k+1} = z_k - a_k g_f(x_{k+1})$, one can easily check that the previous inequality can be rewritten as

$$\begin{aligned} & (A_k + a_k)(f(x_{k+1}) - f(x_\star)) + \frac{1}{2}\|z_{k+1} - x_\star\|_2^2 \\ \leq & A_k(f(x_k) - f(x_\star)) + \frac{1}{2}\|z_k - x_\star\|_2^2 \\ & - \frac{\lambda_k(a_k + A_k) - a_k^2}{2} \|g_f(x_{k+1})\|_2^2, \end{aligned}$$

either by comparing the expressions on a term-by-term basis or by using an appropriate "complete the squares" strategy. We obtain the desired statement by enforcing $\lambda_k(a_k + A_k) - a_k^2 \geq 0$, which allows us to neglect the last term on the right-hand side (which is then nonpositive). Finally, since we have already assumed $a_k \geq 0$, requiring $\lambda_k(a_k + A_k) - a_k^2 \geq 0$ corresponds to

$$0 \leq a_k \leq \frac{\lambda_k + \sqrt{\lambda_k^2 + 4A_k\lambda_k}}{2},$$

which yields the desired result. ■

The convergence speed then follows from the same reasoning as for the proximal point method.

Corollary 5.4. Let $f \in \mathcal{F}_{0,\infty}$, $\{\lambda_i\}_{i \geq 0}$ be a sequence of nonnegative step sizes, and $\{x_i\}_{i \geq 0}$ be the corresponding sequence of iterates from Algorithm 23. For all $k \in \mathbb{N}$, $k \geq 1$, it holds that

$$f(x_k) - f_\star \leq \frac{2\|x_0 - x_\star\|_2^2}{\left(\sum_{i=0}^{k-1} \sqrt{\lambda_i}\right)^2}.$$

Proof. Using the potential from Theorem 5.3:

$$\phi_k \triangleq A_k(f(x_k) - f(x_\star)) + \frac{1}{2}\|z_k - x_\star\|_2^2$$

with $A_0 = 0$ as well as the chaining argument used in Corollary 5.2, we obtain

$$f(x_k) - f_\star \leq \frac{\|x_0 - x_\star\|_2^2}{2A_k}.$$

The desired result then follows from

$$A_{k+1} = A_k + a_k = A_k + \frac{\lambda_k + \sqrt{\lambda_k^2 + 4A_k\lambda_k}}{2} \geq A_k + \frac{\lambda_k}{2} + \sqrt{A_k\lambda_k}.$$

Hence, $A_k \geq \left(\sqrt{A_{k-1}} + \frac{1}{2}\sqrt{\lambda_{k-1}}\right)^2 \geq \frac{1}{4}\left(\sum_{i=0}^{k-1}\sqrt{\lambda_i}\right)^2$. ∎

5.4 Exploiting Strong Convexity

In this section, we provide refined convergence results when the function to be minimized is μ-strongly convex (all results from previous sections can be recovered by setting $\mu = 0$). The algebra is slightly more technical, but the message and techniques are the same. While the proofs in the previous section can be seen as particular cases of the proofs presented below, we detail both versions separately to alleviate the algebraic barrier as much as possible.

Proximal point algorithm under strong convexity

We begin by refining the results on the proximal point algorithm. The same modification to the potential function is used to incorporate acceleration in the sequel.

Theorem 5.5. Let f be a closed, μ-strongly convex and proper function. For any $k \in \mathbb{N}$, $A_k, \lambda_k \geq 0$, any x_k, and $A_{k+1} = A_k(1 + \lambda_k\mu) + \lambda_k$, it holds that

$$A_{k+1}(f(x_{k+1}) - f(x_\star)) + \frac{1 + \mu A_{k+1}}{2}\|x_{k+1} - x_\star\|_2^2 \\ \leq A_k(f(x_k) - f(x_\star)) + \frac{1 + \mu A_k}{2}\|x_k - x_\star\|_2^2,$$

with $x_{k+1} = \operatorname{prox}_{\lambda_k f}(x_k)$.

Proof. We perform a weighted sum of the following valid inequalities, which originate from our assumptions.

- Strong convexity between x_{k+1} and $x_\star$ with weight $A_{k+1} - A_k$:

  $$f(x_\star) \geq f(x_{k+1}) + \langle g_f(x_{k+1}); x_\star - x_{k+1}\rangle + \frac{\mu}{2}\|x_\star - x_{k+1}\|_2^2,$$

  with some $g_f(x_{k+1}) \in \partial f(x_{k+1})$ which we further use below.

- Convexity between x_{k+1} and x_k with weight A_k:

$$f(x_k) \geq f(x_{k+1}) + \langle g_f(x_{k+1}); x_k - x_{k+1} \rangle.$$

By performing the weighted sum of these two inequalities, we obtain the following valid inequality:

$$\begin{aligned} 0 \geq & (A_{k+1} - A_k) \Big[f(x_{k+1}) - f(x_\star) + \langle g_f(x_{k+1}); x_\star - x_{k+1} \rangle \\ & + \frac{\mu}{2} \|x_\star - x_{k+1}\|_2^2 \Big] \\ & + A_k \left[f(x_{k+1}) - f(x_k) + \langle g_f(x_{k+1}); x_k - x_{k+1} \rangle \right]. \end{aligned}$$

By matching the expressions term by term and after substituting $x_{k+1} = x_k - \lambda_k g_f(x_{k+1})$ and $A_{k+1} = A_k(1+\lambda_k\mu)+\lambda_k$, one can check that the previous inequality can be rewritten as

$$\begin{aligned} & A_{k+1}(f(x_{k+1}) - f(x_\star)) + \frac{1 + \mu A_{k+1}}{2} \|x_{k+1} - x_\star\|_2^2 \\ \leq \; & A_k(f(x_k) - f(x_\star)) + \frac{1 + \mu A_k}{2} \|x_k - x_\star\|_2^2 \\ & - \lambda_k \frac{A_k(2 + \lambda_k \mu) + \lambda_k}{2} \|g_f(x_{k+1})\|_2^2. \end{aligned}$$

By neglecting the last term on the right-hand side (which is nonpositive), we reach the desired statement. ■

To obtain the convergence speed guaranteed by the previous potential, we have to characterize the growth rate of A_k again, observing that

$$A_{k+1} \geq A_k(1 + \lambda_k \mu) = \frac{A_k}{1 - \frac{\lambda_k \mu}{1 + \lambda_k \mu}}.$$

The following corollary contains our final worst-case guarantee for the proximal point algorithm, which can be converted to its iteration complexity (details below).

Corollary 5.6. Let f be a closed, μ-strongly convex, and proper function with $\mu \geq 0$, $\{\lambda_i\}_{i \geq 0}$ be a sequence of nonnegative step sizes, and $\{x_i\}_{i \geq 0}$ be the corresponding sequence of iterates from the proximal point algorithm. For all $k \in \mathbb{N}$, $k \geq 1$, it holds that

$$f(x_k) - f_\star \leq \frac{\mu \|x_0 - x_\star\|_2^2}{2(\Pi_{i=0}^{k-1}(1 + \lambda_i \mu) - 1)}.$$

Proof. Note that the recurrence for A_k provided in Theorem 5.5 has a simple solution $A_k = ([\Pi_{i=0}^{k-1}(1 + \lambda_i \mu)] - 1)/\mu$. By combining this with

$$f(x_k) - f_\star \leq \frac{\|x_0 - x_\star\|_2^2}{2A_k},$$

as provided by Theorem 5.5, and $A_0 = 0$, we reach the desired statement. ■

As a particular case, note that we recover the case $\mu = 0$ from the previous corollary since $A_k \to \sum_{i=0}^{k-1} \lambda_i$ when μ goes to zero.

For arriving to the iteration complexity for obtaining an approximate solution x_k satisfying $f(x_k) - f_\star \leq \epsilon$ with constant step sizes $\lambda_i = \lambda$, we use the following sufficient condition due to Corollary 5.6:

$$\frac{\mu\|x_0 - x_\star\|_2^2}{2(1+\lambda\mu)^k} = \left(1 - \frac{\lambda\mu}{1+\lambda\mu}\right)^k \frac{\mu\|x_0 - x_\star\|_2^2}{2} \leq \epsilon \;\Rightarrow\; f(x_k) - f_\star \leq \epsilon.$$

A few algebraic manipulations and taking logarithms allows obtaining the following equivalent sufficient condition

$$k \geq \frac{\log\left(\frac{2\epsilon}{\mu\|x_0 - x_\star\|_2^2}\right)}{\log\left(1 - \frac{\lambda\mu}{1+\lambda\mu}\right)} \;\Rightarrow\; f(x_k) - f_\star \leq \epsilon.$$

Finally, using the bound $\log\left(1 - \frac{1}{x}\right) \leq -\frac{1}{x}$ (for all $x \in (1, \infty)$) we arrive to

$$k \geq \frac{1+\lambda\mu}{\lambda\mu} \log\left(\frac{\mu\|x_0 - x_\star\|_2^2}{2\epsilon}\right) \;\Rightarrow\; f(x_k) - f_\star \leq \epsilon.$$

We conclude that the accuracy ϵ is therefore achieved in $O\left(\frac{1+\lambda\mu}{\lambda\mu} \log \frac{1}{\epsilon}\right)$ iterations of the proximal point algorithm when the step size $\lambda_k = \lambda$ is kept constant. This contrasts with $O\left(\frac{1}{\lambda\epsilon}\right)$ in the non-strongly convex case.

Proximal acceleration and inexactness under strong convexity

To accelerate convergence while exploiting strong convexity, we upgrade Algorithm 23 to Algorithm 24, whose analysis follows the same lines as before. For simplicity, the algorithm is optimized for $\delta = \sqrt{1+\lambda_k\mu}$; it can be slightly improved by exploiting the case $0 \leq \delta < \sqrt{1+\lambda_k\mu}$. This method is a simplified version of the A-HPE method of (Barré *et al.*, 2021, Algorithm 5.1).

Algorithm 24 An inexact accelerated proximal point method

Input: A (μ-strongly) convex function f and an initial point x_0.

1: **Initialize** $z_0 = x_0$ and $A_0 = 0$.
2: **for** $k = 0, \ldots$ **do**
3: Pick $A_{k+1} = A_k + \frac{\lambda_k + 2A_k\lambda_k\mu + \sqrt{4A_k^2\lambda_k\mu(\lambda_k\mu+1) + 4A_k\lambda_k(\lambda_k\mu+1) + \lambda_k^2}}{2}$
4: $y_k = x_k + \frac{(A_{k+1} - A_k)(A_k\mu+1)}{A_{k+1} + 2\mu A_k A_{k+1} - \mu A_k^2}(z_k - x_k)$
5: $x_{k+1} \approx_\delta \operatorname{prox}_{\lambda_k f}(y_k)$ (see Eq.(5.5), for some $\delta \in [0, \sqrt{1+\lambda_k\mu}]$)
6: $z_{k+1} = z_k + \mu\frac{A_{k+1} - A_k}{1+\mu A_{k+1}}(x_{k+1} - z_k) - \frac{A_{k+1} - A_k}{1+\mu A_{k+1}} g_f(x_{k+1})$
7: **end for**

Output: Approximate solution x_{k+1}.

Theorem 5.7. Let f be a closed, μ-strongly convex, and proper function. For any $k \in \mathbb{N}$ and $A_k, \lambda_k \geq 0$, the iterates of Algorithm 24 satisfy

$$A_{k+1}(f(x_{k+1}) - f(x_\star)) + \frac{1+\mu A_{k+1}}{2}\|z_{k+1} - x_\star\|_2^2 \\ \leq A_k(f(x_k) - f(x_\star)) + \frac{1+\mu A_k}{2}\|z_k - x_\star\|_2^2.$$

Proof. We perform a weighted sum of the following valid inequalities, which originate from our assumptions.

- Strong convexity between x_{k+1} and $x_\star$ with weight $A_{k+1} - A_k$:
$$f(x_\star) \geq f(x_{k+1}) + \langle g_f(x_{k+1}); x_\star - x_{k+1}\rangle + \frac{\mu}{2}\|x_\star - x_{k+1}\|_2^2,$$
with some $g_f(x_{k+1}) \in \partial f(x_{k+1})$, where this particular subgradient is used repetitively below.
- Strong convexity between x_{k+1} and x_k with weight A_k
$$f(x_k) \geq f(x_{k+1}) + \langle g_f(x_{k+1}); x_k - x_{k+1}\rangle + \frac{\mu}{2}\|x_k - x_{k+1}\|_2^2.$$
- Error magnitude with weight $\frac{A_{k+1}+2\mu A_k A_{k+1} - \mu A_k^2}{2\lambda_k(1+\mu A_{k+1})}$:
$$\|e_k\|_2^2 \leq (1+\lambda_k\mu)\|x_{k+1} - y_k\|_2^2,$$
which is valid for all $\delta \in [0, \sqrt{1+\lambda_k\mu}]$ in (5.5).

By performing a weighted sum of these three inequalities, with their respective weights, we obtain the following valid inequality:

$$\begin{aligned} 0 \geq &(A_{k+1} - A_k)\Big[f(x_{k+1}) - f(x_\star) + \langle g_f(x_{k+1}); x_\star - x_{k+1}\rangle \\ &\qquad + \frac{\mu}{2}\|x_\star - x_{k+1}\|_2^2\Big] \\ &+ A_k\left[f(x_{k+1}) - f(x_k) + \langle g_f(x_{k+1}); x_k - x_{k+1}\rangle + \frac{\mu}{2}\|x_k - x_{k+1}\|_2^2\right] \\ &+ \frac{A_{k+1} + 2\mu A_k A_{k+1} - \mu A_k^2}{2\lambda_k(1+\mu A_{k+1})}[\|e_k\|_2^2 - (1+\lambda_k\mu)\|x_{k+1} - y_k\|_2^2]. \end{aligned}$$

By matching the expressions term by term and by substituting the expressions for y_k, $x_{k+1} = y_k - \lambda_k g_f(x_{k+1}) + e_k$, and z_{k+1}, one can check that the previous inequality can be rewritten as (we advise against substituting A_{k+1} at this stage):

$$\begin{aligned} &A_{k+1}(f(x_{k+1}) - f(x_\star)) + \frac{1+\mu A_{k+1}}{2}\|z_{k+1} - x_\star\|_2^2 \\ \leq\ &A_k(f(x_k) - f(x_\star)) + \frac{1+\mu A_k}{2}\|z_k - x_\star\|_2^2 \\ &- \frac{(A_{k+1} + 2\mu A_k A_{k+1} - \mu A_k^2)\lambda_k - (A_{k+1} - A_k)^2}{1+\mu A_{k+1}}\,\frac{1}{2}\|g_f(x_{k+1})\|_2^2 \\ &- \frac{A_k(A_{k+1} - A_k)(1+\mu A_k)}{A_{k+1} + 2\mu A_k A_{k+1} - \mu A_k^2}\,\frac{\mu}{2}\|x_k - z_k\|_2^2. \end{aligned}$$

The conclusion follows from $A_{k+1} \geq A_k$ which allows us to discard the last term (which is then nonpositive). Positivity of the first residual term can be enforced by choosing A_{k+1} such that

$$(A_{k+1} + 2\mu A_k A_{k+1} - \mu A_k^2)\lambda_k - (A_{k+1} - A_k)^2 \geq 0.$$

The desired result is achieved by specifically choosing the largest root of the second-order polynomial in A_{k+1}, such that $A_{k+1} \geq A_k$. ■

In contrast with the previous proximal point algorithm, this accelerated version requires

$$O\left(\sqrt{\tfrac{1+\lambda\mu}{\lambda\mu}} \log \tfrac{1}{\epsilon}\right)$$

inexact proximal iterations to reach $f(x_k) - f(x_\star) \leq \epsilon$ when using a constant step size $\lambda_k = \lambda$. This follows from characterizing the growth rate of the sequence $\{A_k\}_k$:

$$A_{k+1} \geq A_k(1 + \lambda_k\mu) + A_k\sqrt{\lambda_k\mu(1 + \lambda_k\mu)} = \frac{A_k}{1 - \sqrt{\frac{\lambda_k\mu}{1+\lambda_k\mu}}}. \tag{5.7}$$

Corollary 5.8. Let $f \in \mathcal{F}_{\mu,\infty}$ with $\mu \geq 0$, $\{\lambda_i\}_{i\geq 0}$ be a sequence of nonnegative step sizes, and $\{x_i\}_{i\geq 0}$ be the corresponding sequence of iterates from Algorithm 24. For all $k \in \mathbb{N}$, $k \geq 1$, it holds that

$$f(x_k) - f_\star \leq \Pi_{i=1}^{k-1}\left(1 - \sqrt{\tfrac{\lambda_i\mu}{1+\lambda_i\mu}}\right)\frac{\|x_0 - x_\star\|_2^2}{2\lambda_0}.$$

Proof. The proof follows from the same arguments as before; that is,

$$f(x_k) - f_\star \leq \frac{\|x_0 - x_\star\|_2^2}{2A_k},$$

and

$$A_k \geq \frac{\lambda_0}{\Pi_{i=1}^{k-1}\left(1 - \sqrt{\frac{\lambda_i\mu}{1+\lambda_i\mu}}\right)},$$

where we used $A_0 = 0$ and $A_1 = \lambda_0$. We then proceed with (5.7). ■

Before continuing to the next section, we note that combining Corollary 5.4 with Corollary 5.8 shows that

$$f(x_k) - f_\star \leq \min\left\{\frac{\Pi_{i=1}^{k-1}\left(1 - \sqrt{\frac{\lambda_i\mu}{1+\lambda_i\mu}}\right)}{\lambda_0}, \frac{4}{\left(\sum_{i=0}^{k-1}\sqrt{\lambda_i}\right)^2}\right\}\frac{\|x_0 - x_\star\|_2^2}{2}.$$

5.5 Application: Catalyst Acceleration

In what follows, we illustrate how to use proximal methods as meta-algorithms to improve the convergence of simple gradient-based first-order methods. The idea consists of using

a base first-order method, such as gradient descent, to obtain approximations to the proximal point subproblems, within an accelerated proximal point method. This idea can be extended by embedding any algorithm that can solve the proximal subproblem.

There exist many notions of *approximate solutions* to the proximal subproblems, giving rise to different types of guarantees together with slightly different methods. In particular, we required the approximate solution to have a small gradient. Other notions of approximate solutions are used, among others, in (Güler, 1992; Schmidt *et al.*, 2011; Villa *et al.*, 2013). Depending on the target application or on the target algorithm for solving inner problems, the *natural* notion of an approximate solution to the proximal subproblem might change. A fairly general framework was developed by Monteiro and Svaiter (2013) (where the error is controlled via a primal-dual gap on the proximal subproblem).

5.5.1 Catalyst acceleration

A popular application of the inexact accelerated proximal gradient is Catalyst acceleration (Lin *et al.*, 2015). For readability purposes, we do not present the general Catalyst framework but rather a simple instance. Stochastic versions of this acceleration procedure have also been developed, and we briefly summarize them in Section 5.5.4. The idea is again to use a base first-order method to approximate the proximal subproblem up to the required accuracy. For now, we assume that we want to minimize an L-smooth convex function f, i.e.,

$$\min_{x \in \mathbb{R}^d} f(x),$$

The corresponding proximal subproblem has the form

$$\operatorname{prox}_{\lambda f}(y) \triangleq \operatorname*{argmin}_x \left(f(x) + \frac{1}{2\lambda} \|x - y\|_2^2 \right), \tag{5.8}$$

and it is therefore the minimization of an $(L + 1/\lambda)$-smooth and $1/\lambda$-strongly convex function. To solve such a problem, one can use a first-order method to approximate its solution.

Preliminaries

In what follows, we consider using a method $\mathcal{M}$ to solve the proximal subproblem (5.8). We assume that this method is guaranteed to converge linearly on any smooth strongly convex problem with minimizer $w_\star$, and more precisely that:

$$\|w_k - w_\star\|_2 \leq C_{\mathcal{M}}(1 - \tau_{\mathcal{M}})^k \|w_0 - w_\star\|_2 \tag{5.9}$$

(where $\{w_i\}_i$ are the iterates of $\mathcal{M}$) for some constant $C_{\mathcal{M}} \geq 0$ and some $0 < \tau_{\mathcal{M}} \leq 1$. Note that we consider linear convergence in terms of $\|w_k - w_\star\|_2$ for convenience; other notions can be used, such as convergence in function values.

We distinguish the sequences $\{x_k\}_k$, $\{y_k\}_k$, and $\{z_k\}_k$, which are the iterates of the inexact accelerated proximal point algorithm (Algorithm 23, or 24), and the sequence of

iterates $\{w_i^{(k)}\}_i$, which are the iterates of $\mathcal{M}$, used to approximate $\text{prox}_{\lambda f}(y_k)$ in step 6 of Algorithm 23 (or step 5 of Algorithm 24). We also use the *warm-start strategy* $w_0^{(k)} = y_k$.

We can thus apply Algorithm 23 to minimize f while (approximately) solving the proximal subproblems with $\mathcal{M}$. We first define four iteration counters:

1. N_{outer}, the number of iterations of the inexact accelerated proximal point method (Algorithm 23, or 24), which serves as the "outer loop" for the overall acceleration scheme. That is, the output of the overall method is $x_{N_{\text{outer}}}$ in the notation of Algorithm 23 (or Algorithm 24);

2. $N_{\text{inner}}(k)$, the number of iterations needed by method $\mathcal{M}$ to approximately solve the proximal subproblem at iteration k of the outer loop of the inexact proximal point method. That is, the number of iterations of $\mathcal{M}$ for approximating $\text{prox}_{\lambda f}(y_k)$ to the target accuracy, when the initial iterate of $\mathcal{M}$ is set to $w_0^{(k)} = y_k$;

3. N_{useless}, the number of iterations performed by $\mathcal{M}$ that did not result in an additional iteration of the inexact accelerated proximal point method. That is, if the user has a limited budget in terms of a total number of iterations for $\mathcal{M}$, it is likely that the last few iterations of $\mathcal{M}$ do not allow completing an iteration of the "outer loop". Thus, $N_{\text{useless}} < N_{\text{inner}}(N_{\text{outer}})$, i.e., the number of useless iterations of $\mathcal{M}$ is smaller than the number of iterations that would have lead to an additional outer iteration.

4. N_{total}, the total number of iterations of method $\mathcal{M}$:

$$N_{\text{total}} = N_{\text{useless}} + \sum_{k=0}^{N_{\text{outer}}-1} N_{\text{inner}}(k).$$

   Again, N_{useless} is the number of iterations of $\mathcal{M}$ that did not allow an additional outer iteration to complete.

Overall complexity

As we detail in the sequel, assuming that $\mathcal{M}$ satisfies (5.9), the overall complexity of the combination of methods is guaranteed to be

$$f(x_{N_{\text{outer}}}) - f_\star = O(N_{\text{total}}^{-2}),$$

where $x_{N_{\text{outer}}}$ is the iterate produced after N_{outer} iterations of the inexact accelerated proximal point method or equivalently, the iterate produced after a total number of iterations N_{total} of method $\mathcal{M}$. More precisely, $x_{N_{\text{outer}}}$ is guaranteed to satisfy

$$f(x_{N_{\text{outer}}}) - f(x_\star) \leq \frac{2\|x_0 - x_\star\|_2^2}{\lambda N_{\text{outer}}^2} \leq \frac{2\|x_0 - x_\star\|_2^2}{\lambda \lfloor B_{\mathcal{M},\lambda}^{-1} N_{\text{total}} \rfloor^2},$$

(the first inequality follows from Corollary 5.4 and the second one from the analysis below) where we used $N_{\text{inner}}(k) \leq B_{\mathcal{M},\lambda}$ for all $k \geq 0$ and hence $\lfloor \frac{N_{\text{total}}}{B_{\mathcal{M},\lambda}} \rfloor \leq N_{\text{outer}}$, where

the constant $B_{\mathcal{M},\lambda}$ depends solely on the choice of λ and on properties of $\mathcal{M}$. This $B_{\mathcal{M},\lambda}$ represents the computational burden of approximately solving one proximal subproblem with $\mathcal{M}$, and it satisfies

$$B_{\mathcal{M},\lambda} \leq \frac{\log(C_{\mathcal{M}}(\lambda L + 2))}{\tau_{\mathcal{M}}} + 1.$$

We provide a few simple examples based on gradient methods for smooth strongly convex minimization. For all these methods, the embedding within the inexact proximal framework yields

$$N_{\text{total}} = O\left(B_{\mathcal{M},\lambda}\sqrt{\tfrac{L\|x_0 - x_\star\|_2^2}{\epsilon}}\right) \tag{5.10}$$

iteration complexity in terms of the total number of calls to $\mathcal{M}$ to find a point that satisfies $f(x_{N_{\text{outer}}}) - f(x_\star) \leq \epsilon$. We can make this bound a bit more explicit depending on the choice of $\mathcal{M}$.

- Let $\mathcal{M}$ be a regular gradient method with step size $1/(L+1/\lambda)$ that we use to solve the proximal subproblem. The method is known to converge linearly with $C_{\mathcal{M}} = 1$ and $\tau_{\mathcal{M}} = \frac{1}{1+\lambda L}$ (the inverse condition ratio for the proximal subproblem), and it produces the accelerated rate in (5.10). Note that directly applying the gradient method to the problem of minimizing f yields a much worse iteration complexity: $O\left(\frac{L\|x_0 - x_\star\|_2^2}{\epsilon}\right)$.
- Let $\mathcal{M}$ be a gradient method including an exact line-search. It is guaranteed to converge linearly with $C_{\mathcal{M}} = \lambda L + 1$ (the condition ratio of the proximal subproblem) and $\tau_{\mathcal{M}} = \frac{2}{2+\lambda L}$. The iteration complexity of applying this steepest descent scheme directly to f is similar to that of vanilla gradient descent. One can also choose λ to avoid having an excessively large $B_{\mathcal{M},\lambda}$; for example, $\lambda = 1/L$.
- Let $\mathcal{M}$ be an accelerated gradient method specifically tailored for smooth strongly convex optimization, such as Nesterov's method with constant momentum; see Algorithm 16. It is guaranteed to converge linearly with $C_{\mathcal{M}} = \lambda L + 1$ and $\tau_{\mathcal{M}} = \sqrt{\frac{1}{1+\lambda L}}$. Although there is no working guarantee for this method on the original minimization problem, if f is not strongly convex, it can still be used to minimize f through the inexact proximal point framework, as proximal subproblems are strongly convex.

To conclude, inexact accelerated proximal schemes produce accelerated rates for vanilla optimization methods that converge linearly for smooth strongly convex minimization. The idea of embedding a simple first-order method within an inexact accelerated scheme can be applied to a large array of settings, including to obtain acceleration in strongly convex problems or for stochastic minimization (see below). However, one should note that practical tuning of the corresponding numerical schemes (and particularly of the step size parameters) critically affects the overall performance, as discussed in, e.g., (Lin *et al.*, 2018). This makes effective implementation somewhat tricky. The analysis of non-convex

settings is beyond the scope of this section, but examples of such results can be found in, e.g., (Paquette *et al.*, 2018).

5.5.2 Detailed Complexity Analysis

Recall that function value accuracies, e.g., in Corollary 5.4 are expressed in terms of outer loop iterations. Therefore, to complete the analysis, we need to answer the following question: given a total budget of N_{total} inner iterations of method $\mathcal{M}$, how many iterations of Algorithm 23, N_{outer}, will we perform in the ideal strategy (in other words, what is $B_{\mathcal{M},\lambda}$)? To answer this question, we start by analyzing the computational cost of solving a single proximal subproblem through $\mathcal{M}$.

Computational cost of inner problems. Let

$$\Phi_k(x) \triangleq f(x) + \frac{1}{2\lambda}\|x - y_k\|_2^2$$

be the objective of the proximal subproblem that we aim to solve at iteration k (line 6 of Algorithm 23) centered at y_k. By construction, $\Phi_k(x)$ is $(L + 1/\lambda)$-smooth and $1/\lambda$-strongly convex. Also denote by $w_0 = y_k$ our (warm-started) initial iterate and by $w_0, w_1, \ldots, w_{N_{\text{inner}}(k)}$ the iterates of $\mathcal{M}$ used to solve $\min_x \Phi_k(x)$ (note that we drop the superscript (k) for readability, avoiding the heavier notation $w_0^{(k)}, w_1^{(k)}, \ldots, w_{N_{\text{inner}}(k)}^{(k)}$). We also denote $w_\star(\Phi_k) \triangleq \operatorname{argmin}_x \Phi_k(x)$.

We need to compute an upper bound on the number of iterations $N_{\text{inner}}(k)$ required to satisfy the error criterion (5.5):

$$\|e_{N_{\text{inner}}(k)}\|_2 = \lambda\|\nabla\Phi_k(w_{N_{\text{inner}}(k)})\|_2 \leq \|w_{N_{\text{inner}}(k)} - w_0\|_2, \tag{5.11}$$

where we denote by $N_{\text{inner}}(k) = \inf\{i : \|\nabla\Phi_k(w_i)\|_2 \leq 1/\lambda\|w_i - w_0\|_2\}$ the index of the first iteration such that (5.11) is satisfied: this is precisely the quantity we want to upper bound. We start with the following observations:

- By $(L + 1/\lambda)$-smoothness of Φ_k, we have

$$\|\nabla\Phi_k(w_i)\|_2 \leq (L + 1/\lambda)\|w_i - w_\star(\Phi_k)\|_2, \tag{5.12}$$

  where $w_\star(\Phi_k)$ is the minimizer of Φ_k.

- The triangle inequality applied to $\|w_0 - w_\star(\Phi_k)\|_2$ implies

$$\|w_0 - w_\star(\Phi_k)\|_2 - \|w_i - w_\star(\Phi_k)\|_2 \leq \|w_0 - w_i\|_2. \tag{5.13}$$

Hence, (5.11) is satisfied if the right-hand side of (5.12) is smaller than the left-hand side of (5.13) divided by λ. Thus, for any i for which we can prove

$$(L + 1/\lambda)\|w_i - w_\star(\Phi_k)\|_2 \leq 1/\lambda(\|w_0 - w_\star(\Phi_k)\|_2 - \|w_\star(\Phi_k) - w_i\|_2),$$

we obtain $N_{\text{inner}}(k) \leq i$. Rephrasing this inequality leads to

$$\|w_i - w_\star(\Phi_k)\|_2 \leq \frac{1}{\lambda L + 2}\|w_0 - w_\star(\Phi_k)\|_2.$$

Therefore, by assumption on $\mathcal{M}$, (5.11) is guaranteed to hold as soon as

$$C_{\mathcal{M}}(1-\tau_{\mathcal{M}})^i \leq \frac{1}{\lambda L + 2},$$

and thus (5.11) holds for any i that satisfies

$$i \geq \left\lceil \frac{\log\left(C_{\mathcal{M}}(\lambda L + 2)\right)}{\log\left(1/(1-\tau_{\mathcal{M}})\right)} \right\rceil.$$

We conclude that (5.11) is satisfied before this number of iterations is achieved; hence,

$$N_{\text{inner}}(k) \leq \left\lceil \frac{\log\left(C_{\mathcal{M}}(\lambda L + 2)\right)}{\log\left(1/(1-\tau_{\mathcal{M}})\right)} \right\rceil \leq \frac{\log\left(C_{\mathcal{M}}(\lambda L + 2)\right)}{\log\left(1/(1-\tau_{\mathcal{M}})\right)} + 1.$$

Given that the right-hand side does not depend on k, we use the notation

$$B_{\mathcal{M},\lambda} \triangleq \frac{\log\left(C_{\mathcal{M}}(\lambda L + 2)\right)}{\log\left(1/(1-\tau_{\mathcal{M}})\right)} + 1$$

as our upper bound on the iteration cost of solving the proximal subproblem via $\mathcal{M}$.

Global complexity bound. We have shown that the number of iterations in the inner loop is bounded above by a constant that depends on the specific choice of the regularization parameter and on the method $\mathcal{M}$. In other words, $N_{\text{inner}}(k) \leq B_{\mathcal{M},\lambda}$. Denoting by N_{total} the total number of calls to the gradient of f, by N_{outer} the number of iterations performed by Algorithm 23, and by N_{useless} the number of iterations of $\mathcal{M}$ that did not result in an additional outer iteration (see discussions in Section 5.5.1 "Preliminaries"), we conclude that

$$N_{\text{total}} = N_{\text{useless}} + \sum_{k=0}^{N_{\text{outer}}-1} N_{\text{inner}}(k) < (N_{\text{outer}} + 1)B_{\mathcal{M},\lambda}.$$

Hence, $N_{\text{outer}} \geq \lfloor B_{\mathcal{M},\lambda}^{-1} N_{\text{total}} \rfloor$ since $N_{\text{useless}} < B_{\mathcal{M},\lambda}$ (the number of useless iterations is smaller than the number of iterations that would lead to an additional outer iteration). The conclusion follows from Corollary 5.4:

$$f(x_{N_{\text{outer}}}) - f(x_\star) \leq \frac{2\|x_0 - x_\star\|_2^2}{\lambda N_{\text{outer}}^2} \leq \frac{2\|x_0 - x_\star\|_2^2}{\lambda \lfloor B_{\mathcal{M},\lambda}^{-1} N_{\text{total}} \rfloor^2}.$$

That is, given a target accuracy ϵ, the iteration complexity written in terms of the total number of approximate proximal minimizations in Algorithm 23 is $O(\sqrt{\frac{\|x_0 - x_\star\|_2^2}{\lambda\epsilon}})$, and the total iteration complexity when solving the problem using $\mathcal{M}$ in the inner loops is simply the same bound multiplied by the cost of solving a single proximal subproblem, namely $O(B_{\mathcal{M},\lambda}\sqrt{\frac{\|x_0 - x_\star\|_2^2}{\lambda\epsilon}})$.

5.5.3 Catalyst for Strongly Convex Problems

The previous analysis holds for the convex (but not necessarily strongly convex) case. The iteration complexity of solving inner problem remains valid in the strongly convex case, and the expression for $B_{\mathcal{M},\lambda}$ can only be improved slightly—by taking into account the better strong convexity parameter $\mu + 1/\lambda$ and the possibly larger acceptable error magnitude with the factor $\sqrt{1+\lambda\mu}$ in Algorithm 24. Therefore, the total number of iterations of Algorithm 24 embedded with $\mathcal{M}$ remains bounded in a similar fashion, and the overall error decreases as $\left(1 - \sqrt{\frac{\lambda\mu}{1+\lambda\mu}}\right)^{\lfloor N_{\text{outer}}/B_{\mathcal{M},\lambda}\rfloor}$, and the iteration complexity is therefore of order

$$O\left(B_{\mathcal{M},\lambda}\sqrt{\frac{1+\lambda\mu}{\lambda\mu}}\log\frac{1}{\epsilon}\right). \tag{5.14}$$

It is thus natural to choose the value of λ by optimizing the overall iteration complexity of Algorithm 24 combined with $\mathcal{M}$. One way to proceed is by optimizing

$$\sqrt{\frac{1+\lambda\mu}{\lambda\mu}}\Big/\tau_{\mathcal{M}}\,,$$

essentially neglecting the factor $\log(C_{\mathcal{M}}(\lambda L+2))$ in the complexity estimate (5.14). Here are a few examples:

- Gradient method with suboptimal tuning (e.g., when using backtracking or line-search techniques): $\tau_{\mathcal{M}} = \frac{\mu\lambda+1}{L\lambda+1}$. Optimizing the ratio leads to the choice $\lambda = \frac{1}{L-2\mu}$, and the ratio is equal to $2\sqrt{\frac{L}{\mu}-1}$. Assuming $C_{\mathcal{M}} = 1$ (which is the case for the standard step size $1/L$), the overall iteration complexity is then $O\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\epsilon}\right)$, where we neglected the factor $\log(2\frac{1-\mu/L}{1-2\mu/L}) \approx \log 2$ when L/μ is large enough.

- Gradient method with optimal tuning: $\tau_{\mathcal{M}} = \frac{2(\mu\lambda+1)}{(L\lambda+\mu\lambda+2)}$. The resulting choice is $\lambda = \frac{2}{L-3\mu}$ and the ratio is $\sqrt{2}\sqrt{\frac{L}{\mu}-1}$, thereby arriving at the same $O\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\epsilon}\right)$.

5.5.4 Catalyst for Randomized/Stochastic Methods

Similar results hold for stochastic methods, assuming the convergence of $\mathcal{M}$ in expectation instead of (5.9), such as in the form $\mathbb{E}\|w_k - w_\star\|_2 \le C_{\mathcal{M}}(1-\tau_{\mathcal{M}})^k\|w_0 - w_\star\|_2$. Overall, the idea remains the same:

1. Use the inexact accelerated proximal point algorithm (Algorithm 23 or 24) as if $\mathcal{M}$ were deterministic.

2. Use the stochastic method $\mathcal{M}$ to obtain points that satisfy the accuracy requirement.

Dealing with the computational burden of solving the inner problem is a bit more technical, but the overall analysis remains similar. One can bound the expected number

of iterations needed to solve the inner problem $\mathbb{E}[N_{\text{inner}}(k)]$ by some constant $B^{(\text{stoch})}_{\mathcal{M},\lambda}$ of the form (details below)

$$B^{(\text{stoch})}_{\mathcal{M},\lambda} \triangleq \frac{\log\left(C_{\mathcal{M}}(\lambda L+2)\right)}{\log\left(1/(1-\tau_{\mathcal{M}})\right)}+2,$$

which is simply $B^{(\text{stoch})}_{\mathcal{M},\lambda} = B_{\mathcal{M},\lambda}+1$. A simple argument for obtaining this bound uses Markov's inequality as follows:

$$\begin{aligned}\mathbb{P}(N_{\text{inner}}(k)>i) &\leq \mathbb{P}\left(\|w_i - w_\star(\Phi_k)\|_2 > \tfrac{1}{\lambda L+2}\|w_0 - w_\star(\Phi_k)\|_2\right)\\ &\leq \frac{\mathbb{E}[\|w_i - w_\star(\Phi_k)\|_2]}{\frac{1}{\lambda L+2}\|w_0 - w_\star(\Phi_k)\|_2} \quad \text{(Markov)}\\ &\leq \frac{C(1-\tau)^i\|w_0 - w_\star(\Phi_k)\|_2}{\frac{1}{\lambda L+2}\|w_0 - w_\star(\Phi_k)\|_2} = \frac{C(1-\tau)^i}{\frac{1}{\lambda L+2}}.\end{aligned}$$

We then use a refined version of this bound: $\mathbb{P}(N_{\text{inner}}(k)>i) \leq \min\left\{1,(\lambda L+2)C(1-\tau)^i\right\}$, and in order to bound $\mathbb{E}[N_{\text{inner}}(k)]$, we proceed with

$$\begin{aligned}\mathbb{E}[N_{\text{inner}}(k)] &= \sum_{t=1}^{\infty}\mathbb{P}(N_{\text{inner}}(k)\geq t)\\ &\leq \int_0^{N_0} 1dt + C(\lambda L+2)\int_{N_0}^{\infty}(1-\tau)^t dt,\end{aligned}$$

where N_0 is such that $1 = C(\lambda L+2)(1-\tau)^{N_0}$. Direct computation yields $\mathbb{E}[N_{\text{inner}}(k)] \leq B^{(\text{stoch})}_{\mathcal{M},\lambda} \triangleq N_0+1$.

The overall expected iteration complexity is that of the inexact accelerated proximal point method multiplied by the expected computational burden of solving the proximal subproblems $B^{(\text{stoch})}_{\mathcal{M},\lambda}$. That is, the expected iteration complexity becomes

$$O\left(B^{(\text{stoch})}_{\mathcal{M},\lambda}\sqrt{\frac{\|x_0-x_\star\|_2^2}{\lambda\epsilon}}\right)$$

in the smooth convex setting, and

$$O\left(B^{(\text{stoch})}_{\mathcal{M},\lambda}\sqrt{\frac{1+\lambda\mu}{\lambda\mu}}\log\frac{1}{\epsilon}\right)$$

in the smooth strongly convex setting. The main argument of this section, namely the use of Markov's inequality, was adapted from Lin *et al.* (2018, Appendix B.4) (merged with the arguments for the deterministic case above). Stochastic versions of Catalyst acceleration were also studied in (Kulunchakov and Mairal, 2019).

5.6 Notes and References

In the optimization literature, the proximal operation is an essential algorithmic primitive at the heart of many practical optimization methods. Proximal point algorithms are also

largely motivated by the fact that they offer a nice framework for obtaining "meta" (or high-level) algorithms. They naturally appear in augmented Lagrangian and splitting-based numerical schemes, among others. We refer the reader to the excellent surveys in (Parikh and Boyd, 2014; Ryu and Boyd, 2016) for more details.

Proximal point algorithms: accelerated and inexact variants. Proximal point algorithms have a long history, dating back to the works of Moreau (1962; 1965): they were introduced to the optimization community by Martinet (1970; 1972). Early interest in proximal methods was motivated by their connection to augmented Lagrangian techniques (Rockafellar, 1973; Rockafellar, 1976; Iusem, 1999); see also the helpful tutorial by Eckstein and Silva (2013)). Among the many other successes and uses of proximal operations, one can cite the many *splitting* techniques (Lions and Mercier, 1979; Eckstein, 1989), for which there are sound surveys (Boyd *et al.*, 2011; Eckstein and Yao, 2012; Condat *et al.*, 2019). In this context, inexact proximal operations had already been introduced by Rockafellar (1976) and were combined with acceleration much later by Güler (1992)—although not with a perfectly rigorous proof, which was later corrected in (Salzo and Villa, 2012; Monteiro and Svaiter, 2013).

Hybrid proximal extragradient (HPE) framework. Whereas Catalyst acceleration is based on the idea of solving the proximal subproblem via a first-order method, the (related) hybrid proximal extragradient framework is also used together with a Newton scheme in (Monteiro and Svaiter, 2013). Furthermore, the accelerated hybrid proximal extragradient framework allows for an increasing sequence of step sizes, thereby leading to faster rates than those obtained via vanilla first-order methods. (That is, using an increasing sequence of $\{\lambda_i\}_i$, $(\sum_{i=1}^N \sqrt{\lambda_i})^2$ might grow much faster than N^2.)

The HPE framework was introduced by Solodov and Svaiter (1999; 1999; 2000; 2001) before it was embedded with acceleration techniques by (Monteiro and Svaiter, 2013).

Catalyst. The variant presented in this section was chosen for simplicity of exposition; it is largely inspired by recent works on the topic in (Lin *et al.*, 2018; Ivanova *et al.*, 2019) along with (Monteiro and Svaiter, 2013). Efficient implementations of Catalyst can be found in the Cyanure package by Mairal (2019). In particular, most efficient practical implementations of Catalyst appear to rely on an *absolute* inaccuracy criterion for the inexact proximal operation, instead of on *relative* (or multiplicative) ones, as used in this section. In practice, the most convenient and efficient variants appear to be those that use a constant number of inner loop iterations to approximately solve the proximal subproblems.

In this section, we chose the relative error model as we believe it allows for a slightly simpler exposition while relying on essentially the same techniques. Catalyst was originally proposed by Lin *et al.* (2015) as a generic tool for reaching accelerated methods. Among others, it allowed for the acceleration of stochastic methods such as SVRG (Johnson and

Zhang, 2013), SAGA (Defazio *et al.*, 2014a), MISO (Mairal, 2015), and Finito (Defazio *et al.*, 2014b) before direct acceleration techniques had been developed for them (Allen-Zhu, 2017; Zhou *et al.*, 2018; Zhou *et al.*, 2019).

Higher-order proximal subproblems. Higher-order proximal subproblems of the form

$$\min_x \left\{ f(x) + \frac{1}{\lambda(p+1)} \|x - x_k\|_2^{p+1} \right\} \tag{5.15}$$

were used by (Nesterov, 2020a; Nesterov, 2020b) as a new primitive for designing optimization schemes. These subproblems can also be solved approximately (via pth-order tensor methods (Nesterov, 2019)) while maintaining good convergence guarantees.

Optimized proximal point methods. It is possible to develop optimized proximal methods in the spirit of optimized gradient methods. That is, given a computational budget—in the proximal setting, this consists of a number of iterations and a sequence of step sizes $\{\lambda_i\}_{0 \leq i \leq N-1}$—one can choose algorithmic parameters to optimize the worst-case performance of a method of the type

$$x_{k+1} = x_0 - \sum_{i=1}^{k} \beta_i g_f(x_i) - \lambda_k g_f(x_{k+1})$$

with respect to the β_i. The proximal equivalent of the optimized gradient method is Güler's second method (Güler, 1992, Section 6), which was obtained as an optimized proximal point method in (Barré *et al.*, 2020a). Alternatively, Güler's second method (Güler, 1992, Section 6) can be obtained by applying the optimized gradient method (without its last iteration trick) to the Moreau envelope of the nonsmooth convex function f. More precisely, denoting by $\tilde{f}$ the Moreau envelope of f, one can apply the optimized gradient method without the last iteration trick to $\tilde{f}$ as $f(x_k) - f_\star = \tilde{f}(x_k) - \tilde{f}_\star - \frac{1}{2L}\|g_{\tilde{f}}(x_k)\|_2^2$, which corresponds precisely to the first term of the potential of the optimized gradient method (see Equation 4.9). In the more general setting of monotone inclusions, one can obtain alternate optimized proximal point methods for different criteria as in (Kim, 2021; Lieder, 2021).

Proofs in this section. The proofs of the potential inequalities in this section were obtained through the performance estimation methodology, introduced by Drori and Teboulle (2014) and specialized to the study of inexact proximal operations by Barré *et al.* (2020a). More details can be found in Section 4.9, "On obtaining the proofs of this section" and in Appendix C. In particular, for reproducibility purposes, we provide code for symbolically verifying the algebraic reformulations of this section at https://github.com/AdrienTaylor/AccelerationMonograph together with those of Section 4.

6

Restart Schemes

In this section, we show that restart strategies can improve the performance of accelerated schemes when the objective function satisfies very generic Hölderian error bounds (HEB) which generalize the notion of strong convexity, but only need to hold locally around the optimum. Restart schemes provide a convenient way to render standard first-order methods adaptive to the HEB parameters, and we will see that the cost of adaptation is only logarithmic.

6.1 Introduction

First-order methods typically exhibit a sublinear convergence, whose rate varies with gradient smoothness. The polynomial upper complexity bounds are typically convex functions of the number of iterations, so first-order methods converge faster in the beginning, then convergence tails off as iterations progress. This suggests that periodically restarting first-order methods, i.e., simply running more "early" iterations, could accelerate their convergence. We illustrate this concept in Figure 6.1.

Beyond this graphical argument, all accelerated methods have memory and look back at least one step to compute the next iterate. They iteratively form a model for the function around the optimum, and restarting allows this model to be periodically refreshed, thereby discarding outdated information as the algorithm converges towards the optimum.

While the benefits of restart are immediately apparent in Figure 6.1, restart schemes raise several important questions: How many iterations should we run between restarts? What is the best complexity bound we can hope for using a restart scheme? What regularity properties of the problem drive the performance of restart schemes? Fortunately, all these questions have an explicit answer that stems from a simple and intuitive argument. We will see that restart schemes are also adaptive to unknown regularity constants and

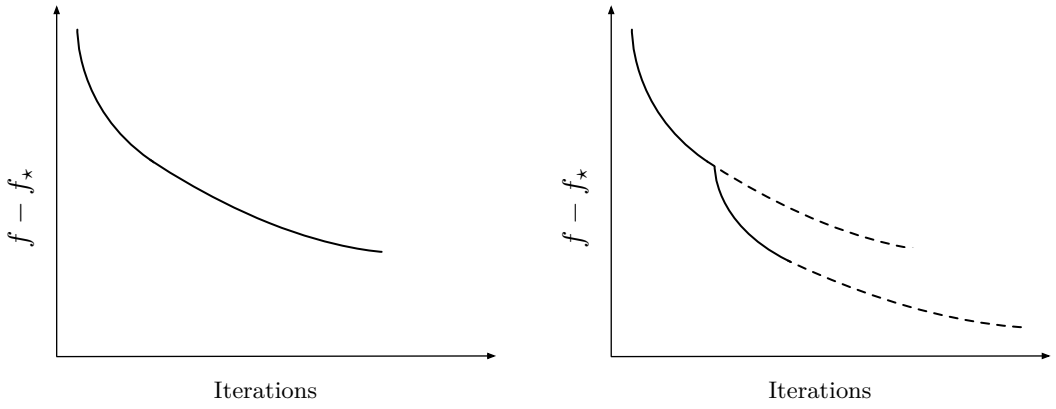

Figure 6.1: *Left:* Sublinear convergence plot without restart. *Right:* Sublinear convergence plot with restart.

often reach near optimal convergence rates without observing these parameters.

We begin by illustrating this adaptivity on the problem of minimizing a strongly convex function using the fixed step gradient method.

6.1.1 The Strongly Convex Case

We illustrate the main argument of this section when minimizing a strongly convex function using fixed step gradient descent. Suppose we seek to solve the minimization problem

$$\min_{x \in \mathbb{R}^d} \; f(x) \tag{6.1}$$

Suppose that the gradient of f is Lipschitz continuous with constant L with respect to the Euclidean norm;

$$\|\nabla f(y) - \nabla f(x)\|_2 \leq L\|y - x\|_2, \quad \text{for all } x, y \in \mathbb{R}^d. \tag{6.2}$$

We can use the fixed step gradient method to solve problem (6.1), as in Algorithm 25 below.

Algorithm 25 Gradient Method

Input: A smooth convex function f and an initial point x_0.

1: **for** $k = 0, \ldots$ **do**

2: $\quad x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$

3: **end for**

Output: An approximate solution x_{k+1}.

The smoothness assumption in (6.2) ensures the complexity bound

$$f(x_k) - f_\star \leq \frac{2L\|x_0 - x_\star\|_2^2}{k + 4} \tag{6.3}$$

after k iterations (see Section 4 for a complete discussion).

Assume now that f is also strongly convex with parameter μ, with respect to the Euclidean norm. Strong convexity means that f satisfies

$$\frac{\mu}{2}\|x - x_\star\|_2^2 \leq f(x) - f_\star, \tag{6.4}$$

where $x_\star$ is an optimal solution to problem (6.1), and $f_\star$ is the corresponding optimal objective value. Denote by $\mathcal{A}(x_0, k)$ the output of k iterations of Algorithm 25 started at x_0, and suppose that we periodically restart the gradient method according to the following scheme.

Algorithm 26 Restart scheme

Input: A smooth convex function f, an initial point x_0 and an inner optimization algorithm $\mathcal{A}(x, k)$.

1: **for** $i = 0, \ldots, N-1$ **do**

2: Obtain x_{i+1} by running k_i iterations of the gradient method, starting at x_i, *i.e.*

$$x_{i+1} = \mathcal{A}(x_i, k_i)$$

3: **end for**

Output: An approximate solution x_N.

Combining the strong convexity bound in (6.4) with the complexity bound in (6.3) yields

$$f(x_{i+1}) - f_\star \leq \frac{2L\|x_i - x_\star\|_2^2}{k+4} \leq \frac{4L}{\mu(k+4)}(f(x_i) - f_\star) \tag{6.5}$$

after an iteration of the restart scheme in Algorithm 26 in which we run k (inner) iterations of the gradient method in Algorithm 25. This means that if we set

$$k_i = k = \left\lceil \frac{8L}{\mu} \right\rceil,$$

then

$$f(x_N) - f_\star \leq \left(\frac{1}{2}\right)^N (f(x_0) - f_\star)$$

after N iterations of the restart scheme in Algorithm 26. Therefore, when running a total of $T = Nk$ gradient steps, we can rewrite the complexity bound in terms of the total number of gradient oracle calls (or inner iterations) as

$$f(x_T) - f_\star \leq \left(\frac{1}{2^{\frac{\mu}{8L}}}\right)^T (f(x_0) - f_\star), \tag{6.6}$$

which proves linear convergence in the strongly convex case.

Of course, the basic gradient method with fixed step size in Algorithm 25 has no memory, so "restarting" it has no impact on the number of iterations or numerical performance. Invoking the restart scheme in Algorithm 26 simply allows us to produce a

better complexity bound in the strongly convex case. Without information about the strong convexity parameter (since restart has no impact on the basic gradient method), whereas the classical bound yields sublinear convergence, while the restart method converges linearly.

Crucially here, the argument in (6.5) can be significantly generalized to improve the convergence rate of several types of first-order methods. In fact, as we will see below, a local bound on the growth rate of the function akin to strong convexity holds almost generically, albeit with a different exponent than in (6.4).

6.1.2 Restart Strategies

Empirical performance of restart schemes was studied at length in (Becker *et al.*, 2011) and various restart strategies were explored to improve convergence of basic gradient methods by exploiting regularity properties of the objective function. (Nesterov, 2013) for example runs a bounded number of iterations between restarts to obtain linear convergence in the strongly convex case, while (O'Donoghue and Candes, 2015) obtain excellent empirical performance by restarting an accelerated method whenever convergence fails to be monotonic (accelerated methods typically exhibit oscillating convergence near the optimum). Below, we will describe the performance of a simple grid search on the restart strategy, attaining optimal performance while using a very limited number of grid points.

6.2 Hölderian Error Bounds

We now recall several results related to subanalytic functions and Hölderian error bounds of the form

$$\frac{\mu}{r} d(x, X_\star)^r \leq f(x) - f_\star, \quad \text{for all } x \in K, \tag{HEB}$$

for some $\mu, r > 0$, where $d(x, X_\star)$ is the distance to the optimal set. We refer the reader to, e.g., (Bolte *et al.*, 2007) for a more complete discussion. These results produce bounds akin to local versions of strong convexity, with various exponents, and they are known to hold under very generic conditions. In general of course, these values are neither observed nor known a priori, but as detailed below, restart schemes can be made adaptive to μ and r and reach optimal convergence rates without any prior information.

6.2.1 Hölderian Error Bound and Smoothness

Let f be a smooth convex function on $\mathbb{R}^d$. Smoothness ensures that

$$f(x) \leq f_\star + \frac{L}{2} \|x - y\|_2^2,$$

for any $x \in \mathbb{R}^d$ and $y \in X_\star$. By setting y to be the projection of x on $X_\star$, this yields the following *upper bound* on suboptimality:

$$f(x) - f_\star \leq \frac{L}{2} d(x, X_\star)^2. \tag{6.7}$$

Now, assume that f satisfies the Hölderian error bound (HEB) on a set K with parameters (r, μ). Combining (6.7) and (HEB) leads to

$$\frac{2\mu}{rL} \leq d(x, X_\star)^{2-r},$$

for every $x \in K$. This means that $2 \leq r$ by taking x close enough to $X_\star$. We will allow the gradient smoothness exponent of 2 to vary in later results, where we assume the gradient to be Hölder smooth, but we first detail the smooth case for simplicity. In what follows, we use the following notations:

$$\kappa \triangleq L/\mu^{\frac{2}{r}} \qquad \text{and} \qquad \tau \triangleq 1 - \frac{2}{r}, \tag{6.8}$$

to define generalized condition numbers for the function f. Note that if $r = 2$, then κ matches the classical condition number of the function.

6.2.2 Subanalytic Functions

Subanalytic functions form a very broad class of functions for which we can demonstrate the Hölderian error bounds as in (HEB), akin to strong convexity. We recall some key definitions and refer the reader to, e.g., (Bolte *et al.*, 2007) for a more complete discussion.

Definition 6.1 (Subanalyticity). (i) A subset $A \subset \mathbb{R}^d$ is called *semianalytic* if each point of $\mathbb{R}^d$ admits a neighborhood V for which $A \cap V$ assumes the following form

$$\bigcup_{i=1}^{p} \bigcap_{j=1}^{q} \{x \in V : f_{ij}(x) = 0, g_{ij}(x) > 0\},$$

where $f_{ij}, g_{ij} : V \to \mathbb{R}$ are real analytic functions for $1 \leq i \leq p$, $1 \leq j \leq q$.
(ii) A subset $A \subset \mathbb{R}^d$ is called *subanalytic* if each point of $\mathbb{R}^d$ admits a neighborhood V such that

$$A \cap V = \{x \in \mathbb{R}^d : (x, y) \in B\}$$

where B is a bounded semianalytic subset of $\mathbb{R}^d \times \mathbb{R}^m$.
(iii) A function $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is called *subanalytic* if its graph is a subanalytic subset of $\mathbb{R}^d \times \mathbb{R}$.

The class of subanalytic functions is, of course, very large, but the definition above suffers from one key shortcoming since the image and preimage of a subanalytic function are not generally subanalytic. To remedy this stability issue, we can define a notion of global subanalyticity. We first define the function β_n with

$$\beta_d(x) \triangleq \left(\frac{x_1}{1 + x_1^2}, \ldots, \frac{x_d}{1 + x_d^2} \right),$$

and we have the following definition.

Definition 6.2 (Global subanalyticity). (i) A subset A of $\mathbb{R}^d$ is called *globally subanalytic* if its image under β_d is a subanalytic subset of $\mathbb{R}^d$.
(ii) A function $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is called *globally subanalytic* if its graph is a globally subanalytic subset of $\mathbb{R}^d \times \mathbb{R}$.

We now recall the Łojasiewicz factorization lemma, which gives us local growth bounds on the graph of a function around its minimum.

Theorem 6.1 (Łojasiewicz factorization lemma). Let $K \subset \mathbb{R}^d$ be a compact set and $g, h : K \to \mathbb{R}$ two continuous globally subanalytic functions. If

$$h^{-1}(0) \subset g^{-1}(0),$$

then

$$\frac{\mu}{r}|g(x)|^r \leq |h(x)|, \quad \text{for all } x \in K, \tag{6.9}$$

for some $\mu, r > 0$.

In an optimization context on a compact set $K \subset \mathbb{R}^d$, we can set $h(x) = f(x) - f_\star$ and $g(x) = d(x, X_\star)$, the Euclidean distance from x to the set $X_\star$, where $X_\star$ is the set of optimal solutions. In this case, we have $h^{-1}(0) \subset g^{-1}(0)$, and we can show that g is globally subanalytic if $X_\star$ is globally subanalytic and f is continuous and globally subanalytic. Theorem 6.1 provides the following Hölderian error bound,

$$\frac{\mu}{r}d(x, X_\star)^r \leq f(x) - f_\star, \quad \text{for all } x \in K, \tag{6.10}$$

for some $\mu, r > 0$. Here, Theorem 6.1 produces a bound on the growth rate of the function around the optimum, generalizing the strong convexity bound in (6.4). We illustrate this in Figure 6.2. Overall, since continuity and subanalyticity are very weak conditions, Theorem 6.1 shows that the Hölderian error bound in (HEB) holds almost generically.

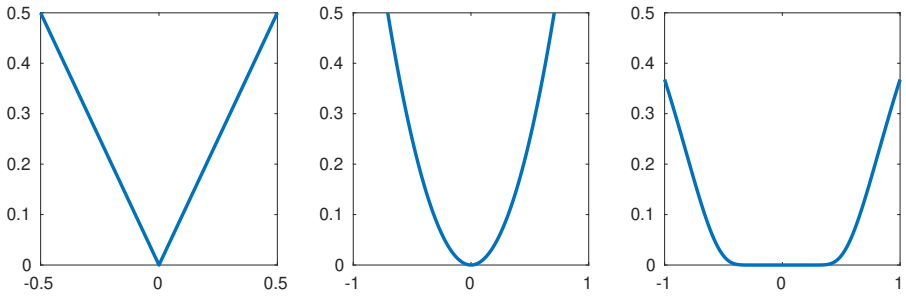

Figure 6.2: *Left and center:* The functions $|x|$ and x^2 satisfy a growth condition around zero. *Right:* The function $\exp(-1/x^2)$ does not.

6.3 Optimal Restart Schemes

We now discuss how the Hölderian error bounds detailed above can be exploited using restart schemes. Generic exponents beyond strong convexity, require restart schemes with a varying number of inner iterations (versus a constant one in the strongly convex case) and we we study here the cost of finding the best such scheme. Suppose again that we seek to solve the following unconstrained minimization problem:

$$\min_{x\in\mathbb{R}^d} \; f(x) \tag{6.11}$$

where the gradient of f is Lipschitz continuous with constant L with respect to the Euclidean norm. The optimal method in (4.9) detailed as Algorithm 11 produces a point x_k that satisfies

$$f(x_k) - f_\star \leq \frac{4L}{k^2}\|x_0 - x_\star\|_2^2 \tag{6.12}$$

after k iterations.

Assuming that the function f satisfies the Hölderian error bound (HEB), we can use a chaining argument similar to that in (6.5) to demonstrate improved convergence rates. While a constant number of inner iterations (between restarts) is optimal in the strongly convex case, the optimal restart scheme for $r > 2$ involves a geometrically increasing number of inner iterations (Nemirovsky and Nesterov, 1985; Roulet and d'Aspremont, 2017).

Theorem 6.2 (Restart complexity)**.** Let f be a smooth convex function satisfying (6.2) with parameter L and (HEB) with parameters (r, μ) on a set K. Assume that we are given $x_0 \in \mathbb{R}^d$ such that $\{x|\, f(x) \leq f(x_0)\} \subset K$. Run the restart scheme in Algorithm 26 from x_0 with iteration the schedule $k_i = C^\star_{\kappa,\tau} e^{\tau i}$, for $i = 1, \dots, R$, where

$$C^\star_{\kappa,\tau} \triangleq e^{1-\tau}(c\kappa)^{\frac{1}{2}}(f(x_0) - f_\star)^{-\frac{\tau}{2}}, \tag{6.13}$$

with κ and τ defined in (6.8) and $c = 4e^{2/e}$. The precision reached at the last point $\hat{x}$ is bounded by,

$$f(\hat{x}) - f_\star \leq \frac{f(x_0) - f_\star}{\left(\tau e^{-1}(f(x_0) - f_\star)^{\frac{\tau}{2}}(c\kappa)^{-\frac{1}{2}}N + 1\right)^{\frac{2}{\tau}}} = O\left(N^{-\frac{2}{\tau}}\right), \tag{6.14}$$

when $\tau > 0$, where $N = \sum_{i=1}^R k_i$ is the total number of inner iterations.

In the strongly convex case, i.e., when $\tau = 0$, the bound above becomes

$$f(\hat{x}) - f_\star \leq \exp\left(-2e^{-1}(c\kappa)^{-\frac{1}{2}}N\right)(f(x_0) - f_\star) = O\left(\exp(-\kappa^{-\frac{1}{2}}N)\right)$$

and we recover the classical linear convergence bound for Algorithm 14 in the strongly convex case. On the other hand, when $0 < \tau < 1$, bound (6.14) reveals a *faster convergence rate than accelerated gradient methods on non-strongly convex functions* (i.e., when $r > 2$).

The closer r is to 2, the tighter the upper and lower bounds induced by smoothness and sharpness are, yielding a better model for the function and faster convergence. This property matches the lower bounds for optimizing smooth sharp functions (Nemirovsky and Nesterov, 1985) up to a constant factor. Moreover, setting $k_i = C^\star_{\kappa,\tau} e^{\tau i}$ yields continuous bounds on the precision, i.e., when $\tau \to 0$, bound (6.14) converges to the linear bound, which shows that for values of τ near zero, constant restart schemes are almost optimal.

6.4 Robustness and Adaptivity

The previous restart schedules depend on the sharpness parameters (r, μ) in (HEB). In general, of course, these values are neither observed nor known a priori. Making the restart scheme adaptive is thus crucial for practical performance. Fortunately, a simple logarithmic grid search on these parameters is enough to guarantee nearly optimal performance. In other words, as shown in (Roulet and d'Aspremont, 2017), the complexity bound in (6.14) is somewhat robust to misspecification of the inner iteration schedule k_i.

6.4.1 Grid Search Complexity

We can test several restart schemes in Algorithm 26, each with a given number of inner iterations N to perform a log-scale grid search on the values of τ and κ in (6.8). We see below that running $(\log_2 N)^2$ restart schemes suffices to achieve nearly optimal performance. We define these schemes as

$$\begin{cases} \mathcal{S}_{p,0} : \text{Restart Algorithm 11 with } k_i = C_p, \\ \mathcal{S}_{p,q} : \text{Restart Algorithm 11 with } k_i = C_p e^{\tau_q i}, \end{cases} \tag{6.15}$$

where $C_p = 2^p$ and $\tau_q = 2^{-q}$. We stop these schemes when the total number of inner algorithm iterations exceeds N, i.e., at the smallest R such that $\sum_{i=1}^R k_i \geq N$. The size of the grid search in C_p is naturally bounded since as we cannot restart the algorithm after more than N total inner iterations, so $p \in [1, \dots, \lfloor \log_2 N \rfloor]$. Also, when τ is smaller than $1/N$, a constant schedule performs as well as the optimal, geometrically increasing schedule, which crucially means we can also choose $q \in [0, \dots, \lceil \log_2 N \rceil]$ and limits the cost of the grid search to $\log_2^2 N$. We have the following complexity bounds.

Theorem 6.3 (Adaptive restart complexity)**.** Let f be a smooth convex function satisfying (6.2) with parameter L and (HEB) with parameters (r, μ) on a set K. Assume that we are given $x_0 \in \mathbb{R}^d$ such that $\{x |\ f(x) \leq f(x_0)\} \subset K$, and let N be a given number of iterations. Run the schemes $\mathcal{S}_{p,q}$, defined in (6.15), for $p \in [1, \dots, \lfloor \log_2 N \rfloor]$ and $q \in [0, \dots, \lceil \log_2 N \rceil]$, stopping each time after N total inner algorithm iterations, i.e., for R such that $\sum_{i=1}^R k_i \geq N$. Assume N is large enough, such that $N \geq 2C^\star_{\kappa,\tau}$, and if $\frac{1}{N} > \tau > 0$, $C^\star_{\kappa,\tau} > 1$.
(i) If $\tau = 0$, there exists $p \in [1, \dots, \lfloor \log_2 N \rfloor]$ such that scheme $\mathcal{S}_{p,0}$ achieves a precision

given by

$$f(\hat{x}) - f_\star \leq \exp\left(-e^{-1}(c\kappa)^{-\frac{1}{2}}N\right)(f(x_0) - f_\star).$$

(ii) If $\tau > 0$, there exist $p \in [1, \ldots, \lfloor \log_2 N \rfloor]$ and $q \in [1, \ldots, \lceil \log_2 N \rceil]$ such that scheme $\mathcal{S}_{p,q}$ achieves a precision given by

$$f(\hat{x}) - f_\star \leq \frac{f(x_0) - f_\star}{\left(\tau e^{-1}(c\kappa)^{-\frac{1}{2}}(f(x_0) - f_\star)^{\frac{\tau}{2}}(N-1)/4 + 1\right)^{\frac{2}{\tau}}}.$$

Overall, running the logarithmic grid search has a complexity that is $(\log_2 N)^2$ times higher than running N iterations using the optimal scheme where we know the parameters in (HEB), while the convergence rate is slowed down by roughly a factor four.

6.5 Extensions

We now discuss several extensions of the results above.

Hölder Smooth Gradient

The results above can be extended somewhat directly to more general notions of regularity. In particular, if we assume that there exist $s \in [1, 2]$ and $L > 0$ on a set $J \subset \mathbb{R}^d$, i.e.,

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2^{s-1}, \quad \text{for all } x, y \in J. \tag{6.16}$$

so that the gradient is Hölder smooth. Without further assumptions on f, the optimal rate of convergence for this class of functions is bounded as $O(1/N^\rho)$, where N is the total number of iterations and

$$\rho = 3s/2 - 1, \tag{6.17}$$

which gives $\rho = 2$ for smooth functions and $\rho = 1/2$ for non-smooth functions. The universal fast gradient method (Nesterov, 2015) achieves this rate. It requires both a target accuracy ϵ and a starting point x_0 as inputs, and it outputs a point $x \triangleq \mathcal{U}(x_0, \epsilon, t)$ such that

$$f(x) - f_\star \leq \frac{\epsilon}{2} + \frac{cL^{\frac{2}{s}}d(x_0, X_\star)^2}{\epsilon^{\frac{2}{s}}t^{\frac{2\rho}{s}}}\frac{\epsilon}{2}, \tag{6.18}$$

after t iterations, where c is a constant ($c = 2^{\frac{4s-2}{s}}$). We can extend the definition of κ and τ in (6.8) to the case where the gradient is Hölder smooth, with

$$\kappa \triangleq \frac{L^{\frac{2}{s}}}{\mu^{\frac{2}{r}}} \qquad \text{and} \qquad \tau \triangleq 1 - \frac{s}{r}. \tag{6.19}$$

We see that τ acts as an analytical condition number that measures the tightness of upper and lower bound models. The key difference from the smooth case described above is that here we need to schedule *both* the target accuracy ϵ_i used by the algorithm *and* the number of iterations k_i made at the i^{th} run of the algorithm. Our scheme is described in Algorithm 27.

Algorithm 27 Universal scheduled restarts for convex minimization

Input: $x_0 \in \mathbb{R}^d$, $\epsilon_0 \geq f(x_0) - f_\star$, $\gamma \geq 0$, a sequence k_i and an inner algorithm $\mathcal{U}(x, \epsilon, k)$.

1: **for** $i = 1, \ldots, R$ **do**
2: $\quad \epsilon_i = e^{-\gamma}\epsilon_{i-1}$
3: $\quad x_i = \mathcal{U}(x_{i-1}, \epsilon_i, k_i)$
4: **end for**

Output: An approximate solution x_R.

We choose a sequence k_i that ensures

$$f(x_i) - f_\star \leq \epsilon_i,$$

for the geometrically decreasing sequence ϵ_i. A grid search on the restart scheme still works in this case, but it requires knowledge of both s and τ.

Theorem 6.4. Let f be a convex function satisfying (6.16) with parameters (s, L) on a set J and (HEB) with parameters (r, μ) on a set K. Given $x_0 \in \mathbb{R}^d$ assume that $\{x | f(x) \leq f(x_0)\} \subset J \cap K$. Run the restart scheme in Algorithm 27 from x_0 for a given $\epsilon_0 \geq f(x_0) - f_\star$ with

$$\gamma = \rho, \qquad k_i = C^\star_{\kappa,\tau,\rho} e^{\tau i}, \quad \text{where} \quad C^\star_{\kappa,\tau,\rho} \triangleq e^{1-\tau}(c\kappa)^{\frac{s}{2\rho}} \epsilon_0^{-\frac{\tau}{\rho}},$$

and where ρ is defined in (6.17), κ and τ are defined in (6.19), and $c = 8e^{2/e}$ here. The precision reached at the last point x_R is given by

$$f(x_R) - f_\star \;\leq\; \exp\left(-\rho e^{-1}(c\kappa)^{-\frac{s}{2\rho}} N\right) \epsilon_0 \;=\; O\left(\exp(-\kappa^{-\frac{s}{2\rho}} N)\right),$$

when $\tau = 0$, whereas when $\tau > 0$,

$$f(x_R) - f_\star \;\leq\; \frac{\epsilon_0}{\left(\tau e^{-1}(c\kappa)^{-\frac{s}{2\rho}} \epsilon_0^{\frac{\tau}{\rho}} N + 1\right)^{-\frac{\rho}{\tau}}} \;=\; O\left(\kappa^{\frac{s}{2\tau}} N^{-\frac{\rho}{\tau}}\right),$$

where $N = \sum_{i=1}^R k_i$ is the total number of iterations.

Relative Smoothness

We can also extend the inequality defining condition (HEB) by replacing the distance to the optimal set by a more general Bregman divergence. Suppose $h(x)$ is a 1-strongly convex function with respect to the Euclidean norm. The Bregman divergence $D_h(x, y)$ is defined as

$$D_h(x, y) \triangleq h(x) - h(y) - \langle \nabla f(y); (x - y) \rangle, \tag{6.20}$$

and we say that a function f is L-smooth with respect to h on $\mathbb{R}^d$ if

$$f(y) \leq f(x) + \langle \nabla f(x); (y - x) \rangle + L D_h(y, x), \quad \text{for all } x, y \in \mathbb{R}^d. \tag{6.21}$$

We can then extend the Hölderian error bound to the Bregman setting as follows. In an optimization context, on a compact set $K \subset \mathbb{R}^d$, we can set $h(x) = f(x) - f_\star$ and $g(x) = D(x, X_\star) = \inf_{y \in X_\star} D_h(x, y)$, where $X_\star$ is the set of optimal solutions. In this case, we have $h^{-1}(0) \subset g^{-1}(0)$, and we can show that g is globally subanalytic if $X_\star$ is subanalytic and if f is continuous and globally subanalytic. Theorem 6.1 shows that

$$\frac{\mu}{r} D(x, X_\star)^r \leq f(x) - f_\star, \quad \text{for all } x \in K, \tag{HEB-B}$$

for some $\mu, r > 0$. This allows us to use the restart scheme complexity results above to accelerate proximal gradient methods.

6.6 Calculus Rules

In general, the exponent r and the factor μ in the bounds (HEB) and (6.25) are not observed and are difficult to estimate. Nevertheless, due to the robustness result in Theorem 6.3, searching for the best restart scheme only introduces a log factor in the overall algorithm complexity. There are, however, a number of scenarios where we can produce much more precise estimates of r and μ and hence both obtain refined a priori complexity bounds and reduce the cost of the grid search in (6.15).

In particular, (Li and Pong, 2018) provides "calculus rules" for the HEB exponent for a number of elementary operations using a related type of error bound known as the Kurdyka-Łojasiewicz inequality; see (Bolte *et al.*, 2007, Theorem 5) for more details on the relationship between these two notions. The results focus on the Kurdyka-Łojasiewicz exponent α, defined as follows.

Definition 6.3. A proper closed convex function has a Kurdyka-Łojasiewicz (KL) exponent α if and only if for any point $\bar{x} \in \mathrm{dom}\partial f$ there is a neighborhood $\mathcal{V}$ of $\bar{x}$, a constant $\nu > 0$, and a constant $c > 0$ such that

$$D(\partial f(x), 0) \geq c(f(x) - f(\bar{x}))^\alpha \tag{6.22}$$

whenever $x \in \mathcal{V}$ and $f(\bar{x}) \leq f(x) \leq f(\bar{x}) + \nu$.

In particular, (Bolte *et al.*, 2007, Theorem 3.3) shows that (HEB) implies (6.22) with exponent $\alpha = 1 - 1/r$. The other way also holds with $r = 1/(1 - \alpha)$, but the constant is degraded; see (Bolte *et al.*, 2007, Section 3.1). Very briefly, the following calculus rules apply to the exponent α.

- If $f(x) = \min_i f_i(x)$ and each f_i has the KL exponent α_i, then f has the KL exponent $\alpha = \max_i \alpha_i$ (Li and Pong, 2018, Corollary 3.1).
- Let $f(x) = g \circ F(x)$, where g is a proper closed function and F is a continuously differentiable mapping. Suppose in addition that g is a KL function with exponent α and that the Jacobian $JF(x)$ is a surjective mapping at some $\bar{x} \in \mathrm{dom}\partial f$. Then f has the KL property at $\bar{x}$ with exponent α (Li and Pong, 2018, Theorem 3.2).

- If $f(x) = \sum_i f_i(x_i)$ and each f_i is continuous and has the KL exponent α_i, then f has KL exponent $\alpha = \max_i \alpha_i$ (Li and Pong, 2018, Corollary 3.3).
- Let f be a proper closed convex function with a KL exponent $\alpha \in [0, 2/3]$. Suppose further that f is continuous on dom∂f. Fix $\lambda > 0$ and consider

$$F_\lambda(X) = \inf_y \left\{ f(y) + \frac{1}{2\lambda}\|x - y\|^2 \right\}.$$

  Then F_λ has the KL exponent $\alpha = \max\left\{\frac{1}{2}, \frac{\alpha}{2-2\alpha}\right\}$ (Li and Pong, 2018, Theorem 3.4).

Note that a related notion of error bound in which the primal gap is replaced by the norm of the proximal step was studied in, e.g., (Pang, 1987; Luo and Tseng, 1992; Tseng, 2010; Zhou and So, 2017).

6.7 Restarting Other First-Order Methods

The restart argument can be readily extended to other optimization methods provided their complexity bound directly depends on some measure of distance to optimality. This is the case for instance for the Frank-Wolfe method, as detailed in (Kerdreux *et al.*, 2019). Suppose that we seek to solve the following constrained optimization problem

$$\min_{x \in \mathcal{C}} f(x). \tag{6.23}$$

The distance to optimality is now measured in terms of the strong Wolfe gap, defined as follows.

Definition 6.4 (Strong Wolfe gap)**.** Let f be a smooth convex function, $\mathcal{C}$ a polytope, and $x \in \mathcal{C}$ be arbitrary. Then the *strong Wolfe gap* $w(x)$ *over* $\mathcal{C}$ is defined as

$$w(x) \triangleq \min_{S \in \mathcal{S}_x} \max_{y \in S, z \in \mathcal{C}} \langle f(x); (y - z) \rangle, \tag{6.24}$$

where $x \in \mathbf{Co}(S)$ and

$$S_x = \{S \subset \mathbf{Ext}(\mathcal{C}),\ \text{finite},\ x \text{ proper combination of elements of } S\},$$

is the set of proper supports of x.

The inequality that plays the role of the Hölderian error bound in (HEB) for the strong Wolfe gap is then written as follows.

Definition 6.5 (Strong Wolfe primal bound)**.** Let K be a compact neighborhood of $X_\star$ in $\mathcal{C}$, where $X_\star$ is the set of solutions of the constrained optimization problem (6.23). A function f satisfies an r-strong Wolfe primal bound on K, if and only if there exists $r \geq 1$ and $\mu > 0$ such that for all $x \in K$

$$f(x) - f_\star \leq \mu w(x)^r, \tag{6.25}$$

where $f_\star$ it the minimum of f.

Notice that this inequality is an upper bound on the primal gap $f(x) - f_\star$, whereas the Hölderian error bound in (HEB) provides a lower bound. This is because the strong Wolfe gap can be understood as a gradient norm, such that (6.25) is a Łojasiewicz inequality as in (Bolte *et al.*, 2007), instead of a direct consequence of the Łojasiewicz factorization lemma as in (HEB) above.

The regularity of f is measured using the *away curvature* as in (Lacoste-Julien and Jaggi, 2015), with

$$C_f^A \triangleq \sup_{\substack{x,s,v\in\mathcal{C}\\ \eta\in[0,1]\\ y=x+\eta(s-v)}} \frac{2}{\eta^2}\big(f(y) - f(x) - \eta\langle \nabla f(x), s - v\rangle\big), \tag{6.26}$$

allowing us to bound the performance the Fractional Away-Step Frank-Wolfe Algorithm in (Kerdreux *et al.*, 2019), as follows.

Theorem 6.5. Let f be a smooth convex function with away curvature C_f^A. Assume the strong Wolfe primal bound in (6.25) holds for some $1 \leq r \leq 2$. Let $\gamma > 0$ and assume $x_0 \in \mathcal{C}$ is such that $e^{-\gamma} w(x_0, \mathcal{S}_0)/2 \leq C_f^A$. With $\gamma_k = \gamma$, the output of the Fractional Away-Step Frank-Wolfe Algorithm satisfies

$$\begin{cases} f(x_T) - f_\star \leq w_0 \dfrac{1}{\left(1+\tilde{T}C_\gamma^r\right)^{\frac{1}{2-r}}} & \text{when } 1 \leq r < 2 \\ f(x_T) - f_\star \leq w_0 \exp\left(-\dfrac{\gamma}{e^{2\gamma}} \dfrac{\tilde{T}}{8C_f^A \mu}\right) & \text{when } r = 2\ , \end{cases} \tag{6.27}$$

after T steps, with $w_0 = w(x_0, \mathcal{S}_0)$, $\tilde{T} \triangleq T - (|\mathcal{S}_0| - |\mathcal{S}_T|)$, and

$$C_\gamma^r \triangleq \frac{e^{\gamma(2-r)} - 1}{8e^{2\gamma} C_f^A \mu w(x_0, \mathcal{S}_0)^{r-2}}. \tag{6.28}$$

This result is similar to that of Theorem 6.4, and it shows that restart yields linear complexity bounds when the exponent in the strong Wolfe primal bound in (6.25) matches that in the curvature (i.e., $r = 2$) and that it yields to improved linear rates when the exponent r satisfies $1 \leq r < 2$. Crucially, the method here is fully adaptive to the error bound parameters, so no prior knowledge of these parameters is required to get the accelerated rates in Theorem 6.5, and no log-scale grid search is required.

6.8 Application: Compressed Sensing

In some applications such as compressed sensing, under some classical assumptions on the problem data, the exponent r is equal to one and the constant μ can be directly computed from quantities controlling recovery performance. In such problems, a single parameter thus controls both signal recovery and computational performance.

Consider, for instance, a sparse recovery problem using the ℓ_1 norm. Given a matrix $A \in \mathbb{R}^{n\times p}$ and observations $b = Ax_\star$ on a signal $x_\star \in \mathbb{R}^p$, recovery is performed by solving

the ℓ_1 minimization program

$$\begin{array}{ll} \text{minimize} & \|x\|_1 \\ \text{subject to} & Ax = b \end{array} \qquad (\ell_1 \text{ recovery})$$

in the variable $x \in \mathbb{R}^p$. A number of conditions on A have been derived to guarantee that (ℓ_1 recovery) recovers the true signal whenever it is sparse enough. Among these, the null space property (see Cohen *et al.*, 2009 and references therein) is defined as follows.

Definition 6.6 (Null space property)**.** The matrix A satisfies the Null Space Property (NSP) *on support* $S \subset \{1, p\}$ with constant $\alpha \geq 1$ if for any $z \in \text{Null}(A) \setminus \{0\}$,

$$\alpha \|z_S\|_1 < \|z_{S^c}\|_1. \qquad \text{(NSP)}$$

The matrix A satisfies the Null Space Property *at order* s with constant $\alpha \geq 1$ if it satisfies it on every support S of cardinality at most s.

The null space property is a necessary and sufficient condition for the convex program (ℓ_1 recovery) to recover all signals up to some sparsity threshold. We have, the following proposition directly linking the null space property and the Hölderian error bound (HEB).

Proposition 6.1. Given a coding matrix $A \in \mathbb{R}^{n \times p}$ satisfying (NSP) at order s with constant $\alpha \geq 1$, if the original signal $x_\star$ is s-sparse, then for any $x \in \mathbb{R}^p$ satisfying $Ax = b$, $x \neq x_\star$, we have

$$\|x\|_1 - \|x_\star\|_1 > \frac{\alpha - 1}{\alpha + 1} \|x - x_\star\|_1. \qquad (6.29)$$

This implies signal recovery, i.e. optimality of $x_\star$ for (ℓ_1 recovery) and the Hölderian error bound (HEB) with $\mu = \frac{\alpha - 1}{\alpha + 1}$.

6.9 Notes and References

The optimal complexity bounds and exponential restart schemes detailed here can be traced back to (Nemirovsky and Nesterov, 1985). Restart schemes were extensively benchmarked in the numerical toolbox TFOCS by (Becker *et al.*, 2011), with a particular focus on compressed sensing applications. The robustness result showing that a log scale grid search produces near optimal complexity bounds is due to (Roulet and d'Aspremont, 2017).

Restart schemes based on the gradient norm as a termination criterion also reach nearly optimal complexity bounds and adapt to strong convexity (Nesterov, 2013) or HEB parameters (Ito and Fukuda, 2021).

Hölderian error bounds for analytic functions can be traced back to the work of Lojasiewicz (1963). They were extended to much broader classes of functions by (Kurdyka, 1998; Bolte *et al.*, 2007). Several examples of problems in signal processing where this condition holds can be found in, e.g., (Zhou *et al.*, 2015; Zhou and So, 2017). Calculus rules for the exponent are discussed in details in, e.g., (Li and Pong, 2018).

Restarting is also helpful in the stochastic setting, with (Davis *et al.*, 2019) showing recently that stochastic algorithms with geometric step decay converge linearly on functions satisfying Hölderian error bounds. This validates a classical empirical acceleration trick, which is to restarts every few epochs after adjusting the step size (aka the learning rate in machine learning terminology).

Appendices

A

Useful Inequalities

In this appendix, we prove basic inequalities involving smooth strongly convex functions. Most of these inequalities are not used in our developments. Nevertheless, we believe they are useful for gaining intuition about smooth strongly convex of functions, as well as for comparisons with the literature.

Also note that these inequalities can be considered standard (see, e.g., (Nesterov, 2003, Theorem 2.1.5).

A.1 Smoothness and Strong Convexity in Euclidean spaces

In this section, we consider a Euclidean setting, where $\|x\|_2^2 = \langle x; x\rangle$ and $\langle .; .\rangle : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a dot product.

The following theorem summarizes known inequalities that characterize the class of smooth convex functions. Note that these characterizations of $f \in \mathcal{F}_{0,L}$ are all equivalent assuming that $f \in \mathcal{F}_{0,\infty}$ since convexity is not implied by some of the points below. In particular, (i), (ii), (v), (vi), and (vii) do not encode the convexity of f when taken on their own, whereas (iii) and (iv) encode both smoothness and convexity.

Theorem A.1. Let $f : \mathbb{R}^d \to \mathbb{R}$ be a differentiable convex function. The following statements are equivalent for inclusion in $\mathcal{F}_{0,L}$.

(i) ∇f satisfies a Lipschitz condition: for all $x, y \in \mathbb{R}^d$,

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2.$$

(ii) f is upper bounded by quadratic functions: for all $x, y \in \mathbb{R}^d$,

$$f(x) \leq f(y) + \langle \nabla f(y); x - y\rangle + \frac{L}{2}\|x - y\|_2^2.$$

(iii) f satisfies, for all $x, y \in \mathbb{R}^d$,

$$f(x) \geq f(y) + \langle \nabla f(y); x - y \rangle + \frac{1}{2L} \|\nabla f(x) - \nabla f(y)\|_2^2.$$

(iv) ∇f is cocoercive: for all $x, y \in \mathbb{R}^d$,

$$\langle \nabla f(x) - \nabla f(y); x - y \rangle \geq \frac{1}{L} \|\nabla f(x) - \nabla f(y)\|_2^2.$$

(v) ∇f satisfies, for all $x, y \in \mathbb{R}^d$,

$$\langle \nabla f(x) - \nabla f(y); x - y \rangle \leq L \|x - y\|_2^2.$$

(vi) $\frac{L}{2}\|x\|_2^2 - f(x)$ is convex.

(vii) f satisfies, for all $\lambda \in [0, 1]$,

$$f(\lambda x + (1 - \lambda) y) \geq \lambda f(x) + (1 - \lambda) f(y) - \lambda(1 - \lambda) \frac{L}{2} \|x - y\|_2^2.$$

Proof. We start with (i)⇒(ii). We use the first-order expansion

$$f(y) = f(x) + \int_0^1 \langle \nabla f(x + \tau(y - x)); y - x \rangle d\tau.$$

The quadratic upper bound then follows from algebraic manipulations and from upper bounding the integral term:

$$\begin{aligned}
f(y) &= f(x) + \langle \nabla f(x); y - x \rangle \\
&\quad + \int_0^1 \langle \nabla f(x + \tau(y - x)) - \nabla f(x); y - x \rangle d\tau \\
&\leq f(x) + \langle \nabla f(x); y - x \rangle \\
&\quad + \int_0^1 \|\nabla f(x + \tau(y - x)) - \nabla f(x)\|_2 \|y - x\|_2 d\tau \\
&\leq f(x) + \langle \nabla f(x); y - x \rangle + L\|x - y\|_2^2 \int_0^1 \tau d\tau \\
&= f(x) + \langle \nabla f(x); y - x \rangle + \frac{L}{2} \|x - y\|_2^2.
\end{aligned}$$

We proceed with (ii)⇒(iii). The idea is to require the quadratic upper bound to be everywhere above the linear lower bound arising from the convexity of f. That is, for all $x, y, z \in \mathbb{R}^d$,

$$f(y) + \langle \nabla f(y); z - y \rangle \leq f(z) \leq f(x) + \langle \nabla f(x); z - x \rangle + \frac{L}{2} \|x - z\|_2^2.$$

In other words, for all $z \in \mathbb{R}^d$, we must have

$$\begin{aligned} f(y) + \langle \nabla f(y); z - y \rangle &\leq f(x) + \langle \nabla f(x); z - x \rangle + \frac{L}{2}\|x - z\|_2^2 \\ \Leftrightarrow f(y) - f(x) + \langle \nabla f(y); z - y \rangle - \langle \nabla f(x); z - x \rangle - \frac{L}{2}\|x - z\|_2^2 &\leq 0 \\ \Leftrightarrow f(y) - f(x) + \max_{z \in \mathbb{R}^d} \langle \nabla f(y); z - y \rangle - \langle \nabla f(x); z - x \rangle - \frac{L}{2}\|x - z\|_2^2 &\leq 0 \\ \Leftrightarrow f(y) - f(x) + \langle \nabla f(y); x - y \rangle + \frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|_2^2 &\leq 0, \end{aligned}$$

where the last line follows from the explicit maximization on z. That is, we pick $z = x - \frac{1}{L}(\nabla f(x) - \nabla f(y))$ and reach the desired result after base algebraic manipulations.

We continue with (iii)⇒(iv), which simply follows from adding

$$\begin{aligned} f(x) &\geq f(y) + \langle \nabla f(y); x - y \rangle + \frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|_2^2 \\ f(y) &\geq f(x) + \langle \nabla f(x); y - x \rangle + \frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|_2^2. \end{aligned}$$

To obtain (iv)⇒(i), one can use Cauchy-Schwartz:

$$\begin{aligned} \frac{1}{L}\|\nabla f(x) - \nabla f(y)\|_2^2 &\leq \langle \nabla f(x) - \nabla f(y); x - y \rangle \\ &\leq \|\nabla f(x) - \nabla f(y)\|_2 \|x - y\|_2, \end{aligned}$$

which allows us to conclude that $\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2$, thus reaching the final statement.

To obtain (ii)⇒(v), we simply add

$$\begin{aligned} f(x) &\leq f(y) + \langle \nabla f(y); x - y \rangle + \frac{L}{2}\|x - y\|_2^2 \\ f(y) &\leq f(x) + \langle \nabla f(x); y - x \rangle + \frac{L}{2}\|x - y\|_2^2 \end{aligned}$$

and reorganize the resulting inequality.

To obtain (v)⇒(ii), we again use a first-order expansion:

$$f(y) = f(x) + \int_0^1 \langle \nabla f(x + \tau(y - x)); y - x \rangle d\tau.$$

The quadratic upper bound then follows from algebraic manipulations and from upper bounding the integral term. (We use the intermediate variable $z_\tau = x + \tau(y - x)$ for

convenience)

$$\begin{aligned}
f(y) &= f(x) + \langle \nabla f(x); y - x \rangle \\
&\quad + \int_0^1 \langle \nabla f(x + \tau(y - x)) - \nabla f(x); y - x \rangle d\tau \\
&= f(x) + \langle \nabla f(x); y - x \rangle + \int_0^1 \frac{1}{\tau} \langle \nabla f(z_\tau) - \nabla f(x); z_\tau - x \rangle d\tau \\
&\leq f(x) + \langle \nabla f(x); y - x \rangle + \int_0^1 \frac{L}{\tau} \|z_\tau - x\|_2^2 d\tau \\
&= f(x) + \langle \nabla f(x); y - x \rangle + L\|x - y\|_2^2 \int_0^1 \tau d\tau \\
&= f(x) + \langle \nabla f(x); y - x \rangle + \frac{L}{2}\|x - y\|_2^2.
\end{aligned}$$

For the equivalence (vi)⇔(ii), simply define $h(x) = \frac{L}{2}\|x\|_2^2 - f(x)$ (and hence $\nabla h(x) = Lx - \nabla f(x)$) and observe that for all $x, y \in \mathbb{R}^d$,

$$h(x) \geq h(y) + \langle \nabla h(y); x - y \rangle \Leftrightarrow f(x) \leq f(y) + \langle \nabla f(y); x - y \rangle + \frac{L}{2}\|x - y\|_2^2,$$

which follows from base algebraic manipulations.

Finally, the equivalence (vi)⇔(vii) follows the same $h(x) = \frac{L}{2}\|x\|_2^2 - f(x)$ (and hence $\nabla h(x) = Lx - \nabla f(x)$) and the observation that for all $x, y \in \mathbb{R}^d$ and $\lambda \in [0, 1]$, we have

$$\begin{aligned}
&h(\lambda x + (1 - \lambda)y) \leq \lambda h(x) + (1 - \lambda)h(y) \\
&\qquad\qquad \Leftrightarrow \\
&f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda)f(y) - \lambda(1 - \lambda)\frac{L}{2}\|x - y\|_2^2,
\end{aligned}$$

which follows from base algebraic manipulations. ■

To obtain the corresponding inequalities in the strongly convex case, one can rely on Fenchel conjugation between smoothness and strong convexity; see, for example, (Rockafellar and Wets, 2009, Proposition 12.6). The following inequalities are stated without proofs; they can be obtained either as direct consequences of the definitions or from Fenchel conjugation along with the statements of Theorem A.1.

Theorem A.2. Let $f : \mathbb{R}^d \to \mathbb{R}$ be a closed convex proper function. The following statements are equivalent for inclusion in $\mathcal{F}_{\mu,L}$.

(i) ∇f satisfies a Lipschitz and an inverse Lipschitz condition: for all $x, y \in \mathbb{R}^d$,

$$\mu\|x - y\|_2 \leq \|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2.$$

(ii) f is lower and upper bounded by quadratic functions: for all $x, y \in \mathbb{R}^d$,

$$\begin{aligned}
f(y) + \langle \nabla f(y); x - y \rangle + \frac{\mu}{2}\|x - y\|_2^2 \\
\leq f(x) \leq f(y) + \langle \nabla f(y); x - y \rangle + \frac{L}{2}\|x - y\|_2^2.
\end{aligned}$$

(iii) f satisfies, for all $x, y \in \mathbb{R}^d$,

$$\begin{aligned} f(y)+\langle \nabla f(y); x-y\rangle + \frac{1}{2L}\|\nabla f(x)-\nabla f(y)\|_2^2 \\ \leq f(x) \leq \\ f(y) + \langle \nabla f(y); x-y\rangle + \frac{1}{2\mu}\|\nabla f(x)-\nabla f(y)\|_2^2. \end{aligned}$$

(iv) ∇f satisfies, for all $x, y \in \mathbb{R}^d$,

$$\begin{aligned} &\frac{1}{L}\|\nabla f(x)-\nabla f(y)\|_2^2 \\ &\quad \leq \langle \nabla f(x)-\nabla f(y); x-y\rangle \leq \frac{1}{\mu}\|\nabla f(x)-\nabla f(y)\|_2^2. \end{aligned}$$

(v) ∇f satisfies, for all $x, y \in \mathbb{R}^d$,

$$\mu\|x-y\|_2^2 \leq \langle \nabla f(x)-\nabla f(y); x-y\rangle \leq L\|x-y\|_2^2.$$

(vi) For all $\lambda \in [0,1]$,

$$\begin{aligned} \lambda f(x)+(1-\lambda)f(y) - \lambda(1-\lambda)\frac{L}{2}\|x-y\|_2^2 \\ \leq f(\lambda x + (1-\lambda)y) \leq \\ \lambda f(x)+(1-\lambda)f(y) - \lambda(1-\lambda)\frac{\mu}{2}\|x-y\|_2^2. \end{aligned}$$

(vii) $f(x) - \frac{\mu}{2}\|x\|_2^2$ and $\frac{L}{2}\|x\|_2^2 - f(x)$ are convex and $(L-\mu)$-smooth.

Finally, we mention that the existence of an inequality that allows us to encode both smoothness and strong convexity together. This inequality is also known as an *interpolation* inequality (Taylor *et al.*, 2017c), and it turns out to be particularly useful for proving worst-case guarantees.

Theorem A.3. Let $f : \mathbb{R}^d \to \mathbb{R}$ be a differentiable function. f is L-smooth μ-strongly convex if and only if

$$\begin{aligned} f(x) \geq f(y)+\langle \nabla f(y); x-y\rangle + \frac{1}{2L}\|\nabla f(x)-\nabla f(y)\|_2^2 \\ + \frac{\mu}{2(1-\mu/L)}\|x-y-\frac{1}{L}(\nabla f(x)-\nabla f(y))\|_2^2. \end{aligned} \tag{A.1}$$

Proof. ($f \in \mathcal{F}_{\mu,L} \Rightarrow$ (A.1)) The idea is to require the quadratic upper bound from smoothness to be everywhere above the quadratic lower bound arising from strong convexity. That is, for all $x, y, z \in \mathbb{R}^d$

$$\begin{aligned} f(y) + \langle \nabla f(y); z-y\rangle + \frac{\mu}{2}\|z-y\|_2^2 \leq f(z) \leq & f(x) + \langle \nabla f(x); z-x\rangle \\ & + \frac{L}{2}\|x-z\|_2^2. \end{aligned}$$

In other words, for all $z \in \mathbb{R}^d$, we must have

$$\begin{aligned} f(y)+\langle \nabla f(y); z-y\rangle + \frac{\mu}{2}\|z-y\|_2^2 \leq f(x) \\ + \langle \nabla f(x); z-x\rangle + \frac{L}{2}\|x-z\|_2^2 \\ \Leftrightarrow f(y) - f(x) + \langle \nabla f(y); z-y\rangle + \frac{\mu}{2}\|z-y\|_2^2 - \langle \nabla f(x); z-x\rangle \\ - \frac{L}{2}\|x-z\|_2^2 \leq 0 \\ \Leftrightarrow f(y) - f(x) + \max_{z\in\mathbb{R}^d}\left(\langle \nabla f(y); z-y\rangle + \frac{\mu}{2}\|z-y\|_2^2 \right. \\ \left. - \langle \nabla f(x); z-x\rangle - \frac{L}{2}\|x-z\|_2^2\right) \leq 0 \end{aligned}$$

explicit maximization over z. That is, picking $z = \frac{Lx-\mu y}{L-\mu} - \frac{1}{L-\mu}(\nabla f(x) - \nabla f(y))$ allows the desired inequality to be reached by base algebraic manipulations.

((A.1)$\Rightarrow$ $f \in \mathcal{F}_{\mu,L}$) $f \in \mathcal{F}_{0,L}$ is direct by observing that (A.1) is stronger than Theorem A.1(iii); $f \in \mathcal{F}_{\mu,L}$ is then direct by reformulating (A.1) as

$$\begin{aligned} f(x) \geq f(y)+\langle \nabla f(y); x-y\rangle + \frac{\mu}{2}\|x-y\|_2^2 \\ + \frac{1}{2L(1-\mu/L)}\|\nabla f(x) - \nabla f(y) - \mu(x-y)\|_2^2, \end{aligned}$$

which is stronger than $f(x) \geq f(y) + \langle \nabla f(y); x-y\rangle + \frac{\mu}{2}\|x-y\|_2^2$. ■

Remark A.1. It is crucial to recall that some of the inequalities above are only valid when $\mathbf{dom}\, f = \mathbb{R}^d$—in particular, this holds for Theorem A.1(iii & iv), Theorem A.2(iii&iv), and Theorem A.3. We refer to (Drori, 2018) for an illustration that some inequalities are not valid when restricted on some $\mathbf{dom}\, f \neq \mathbb{R}^d$. Most standard inequalities, however, do hold even in the case of restricted domains, as established in, e.g., (Nesterov, 2003). Some other inequalities, such as Theorem A.1(iv) and Theorem A.2(iv), do hold under the additional assumption of twice continuous differentiability(see, for example, (De Klerk *et al.*, 2020)).

A.2 Smoothness for General Norms and Restricted Sets

In this section, we show that requiring a Lipschitz condition on ∇f, on a convex set $C \subseteq \mathbb{R}^d$, implies a quadratic upper bound on f. That is, requiring that for all $x, y \in C$,

$$\|\nabla f(x) - \nabla f(y)\|_* \leq L\|x-y\|,$$

where $\|.\|$ is some norm and $\|.\|_*$ is the corresponding dual norm, implies a quadratic upper bound $\forall x, y \in C$:

$$f(x) \leq f(y) + \langle \nabla f(y); x-y\rangle + \frac{L}{2}\|x-y\|^2.$$

Theorem A.4. Let $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ be continuously differentiable on some open convex set $C \subseteq \mathbb{R}^d$, and let it satisfy a Lipschitz condition

$$\|\nabla f(x) - \nabla f(y)\|_* \leq L\|x - y\|,$$

for all $x, y \in C$. Then, it holds that

$$f(x) \leq f(y) + \langle \nabla f(y); x - y \rangle + \frac{L}{2}\|x - y\|^2,$$

for all $x, y \in C$.

Proof. The desired result is obtained from a first-order expansion:

$$f(y) = f(x) + \int_0^1 \langle \nabla f(x + \tau(y - x)); y - x \rangle d\tau.$$

The quadratic upper bound then follows from algebraic manipulations and from upper bounding the integral term

$$\begin{aligned}
f(y) &= f(x) + \langle \nabla f(x); y - x \rangle \\
&\quad + \int_0^1 \langle \nabla f(x + \tau(y - x)) - \nabla f(x); y - x \rangle d\tau \\
&\leq f(x) + \langle \nabla f(x); y - x \rangle \\
&\quad + \int_0^1 \|\nabla f(x + \tau(y - x)) - \nabla f(x)\|_* \|y - x\| d\tau \\
&\leq f(x) + \langle \nabla f(x); y - x \rangle + L\|x - y\|^2 \int_0^1 \tau d\tau \\
&= f(x) + \langle \nabla f(x); y - x \rangle + \frac{L}{2}\|x - y\|^2. \quad \blacksquare
\end{aligned}$$

B

Variations on Nesterov Acceleration

B.1 Relations between Acceleration Methods

B.1.1 Optimized Gradient Method: Forms I & II

In this short section, we show that Algorithm 9 and Algorithm 10 generate the same sequence $\{y_k\}_k$. A direct consequence of this statement is that the sequences $\{x_k\}_k$ also match, as in both cases they are generated from simple gradient steps on $\{y_k\}_k$.

For this purpose we show that Algorithm 10 is a reformulation of Algorithm 9.

Proposition B.1. The sequence $\{y_k\}_k$ generated by Algorithm 9 is equal to that generated by Algorithm 10.

Proof. We first observe that the sequences are initiated the same way in both formulations of the OGM. Furthermore, consider one iteration of the OGM in form I:

$$y_k = \left(1 - \frac{1}{\theta_{k,N}}\right) x_k + \frac{1}{\theta_{k,N}} z_k.$$

Therefore, we clearly have $z_k = \theta_{k,N} y_k + (1 - \theta_{k,N}) x_k$. At the next iteration, we have

$$\begin{aligned} y_{k+1} &= \left(1 - \frac{1}{\theta_{k+1,N}}\right) x_{k+1} + \frac{1}{\theta_{k+1,N}} \left(z_k - \frac{2\theta_{k,N}}{L} \nabla f(y_k)\right) \\ &= \left(1 - \frac{1}{\theta_{k+1,N}}\right) x_{k+1} \\ &\quad + \frac{1}{\theta_{k+1,N}} \left(\theta_{k,N} y_k + (1 - \theta_{k,N}) x_k - \frac{2\theta_{k,N}}{L} \nabla f(y_k)\right), \end{aligned}$$

where we substituted z_k by its equivalent expression from the previous iteration. Now,

by noting that $-\frac{1}{L}\nabla f(y_k) = x_{k+1} - y_k$, we reach

$$\begin{aligned} y_{k+1} &= \frac{\theta_{k+1,N} - 1}{\theta_{k+1,N}} x_{k+1} + \frac{1}{\theta_{k+1,N}} \left((1 - \theta_{k,N}) x_k + 2\theta_{k,N} x_{k+1} - \theta_{k,N} y_k \right) \\ &= x_{k+1} + \frac{\theta_{k,N} - 1}{\theta_{k+1,N}} (x_{k+1} - x_k) + \frac{\theta_{k,N}}{\theta_{k+1,N}} (x_{k+1} - y_k), \end{aligned}$$

where we reorganized the terms to achieve the same format as in Algorithm 10. ■

B.1.2 Nesterov's Method: Forms I, II, and III

Proposition B.2. The two sequences $\{x_k\}_k$ and $\{y_k\}_k$ generated by Algorithm 11 are equal to those generated by Algorithm 12.

Proof. In order to prove the result, we use the identities $A_{k+1} = a_k^2$ as well as $A_k = \sum_{i=0}^{k-1} a_i$, and $a_{k+1}^2 = a_k^2 + a_{k+1}$.

Given that the sequences $\{x_k\}_k$ are obtained from gradient steps on y_k in both formulations, it is sufficient to prove that the sequences $\{y_k\}_k$ match. The equivalence is clear for $k = 0$, as both methods generate $y_1 = x_0 - \frac{1}{L}\nabla f(x_0)$. For $k \geq 0$, from Algorithm 11, one can write iteration k as

$$y_k = \frac{A_k}{A_{k+1}} x_k + \left(1 - \frac{A_k}{A_{k+1}}\right) z_k,$$

and hence,

$$\begin{aligned} z_k &= \frac{A_{k+1}}{A_{k+1} - A_k} y_k + \left(1 - \frac{A_{k+1}}{A_{k+1} - A_k}\right) x_k \\ &= a_k y_k + (1 - a_k)\, x_k. \end{aligned}$$

Substituting this expression in that for iteration $k+1$, we reach

$$\begin{aligned} y_{k+1} =& \frac{A_{k+1}}{A_{k+2}} x_{k+1} + \frac{A_{k+2} - A_{k+1}}{A_{k+2}} \left(z_k - \frac{A_{k+1} - A_k}{L} \nabla f(y_k) \right) \\ =& \frac{a_k^2}{a_{k+1}^2} x_{k+1} + \frac{1}{a_{k+1}} \left(a_k y_k + (1 - a_k)\, x_k - \frac{a_k}{L} \nabla f(y_k) \right) \\ =& \frac{a_k^2}{a_{k+1}^2} x_{k+1} + \frac{1}{a_{k+1}} \left(a_k x_{k+1} + (1 - a_k)\, x_k \right) \\ =& x_{k+1} + \frac{a_k - 1}{a_{k+1}} (x_{k+1} - x_k), \end{aligned}$$

where we substituted the expression for z_k and used previous identities to reach the desired statement. ■

The same relationship holds with Algorithm 13, as provided by the next proposition.

Proposition B.3. The three sequences $\{z_k\}_k$, $\{x_k\}_k$ and $\{y_k\}_k$ generated by Algorithm 11 are equal to those generated by Algorithm 13.

Proof. Clearly, we have $x_0 = z_0 = y_0$ in both methods. Let us assume that the sequences match up to iteration k, that is, up to y_{k-1}, x_k, and z_k. Clearly, both y_k and z_{k+1} are computed in the same way in both methods. It remains to compare the update rules for x_{k+1}: in Algorithm 13, we have

$$\begin{aligned} x_{k+1} &= \frac{A_k}{A_{k+1}} x_k + \left(1 - \frac{A_k}{A_{k+1}}\right) z_{k+1} \\ &= y_k - \left(1 - \frac{A_k}{A_{k+1}}\right) \frac{A_{k+1} - A_k}{L} \nabla f(y_k), \end{aligned}$$

where we used the update rule for z_{k+1}. Further simplifications, along with the identity $(A_{k+1} - A_k)^2 = A_{k+1}$ allows us to arrive at

$$\begin{aligned} x_{k+1} &= y_k - \frac{(A_{k+1} - A_k)^2}{L A_{k+1}} \nabla f(y_k) \\ &= y_k - \frac{1}{L} \nabla f(y_k), \end{aligned}$$

which is clearly the same update rule as that of Algorithm 11. Hence, all sequences match and the desired statement is proved. ■

B.1.3 Nesterov's Accelerated Gradient Method (Strongly Convex Case): Forms I, II, and III

In this short section, we provide alternate, equivalent, formulations for Algorithm 14.

Algorithm 28 Nesterov's method, form II

Input: L-smooth μ-strongly convex function f and initial point x_0.

1: **Initialize** $z_0 = x_0$; $q = \mu/L$, $A_0 = 0$, and $A_1 = (1-q)^{-1}$.
2: **for** $k = 0, \ldots$ **do**
3: $\quad A_{k+2} = \frac{2A_{k+1}+1+\sqrt{4A_{k+1}+4qA_{k+1}^2+1}}{2(1-q)}$
4: $\quad x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k)$
5: $\quad y_{k+1} = x_{k+1} + \beta_k(x_{k+1} - x_k)$
6: $\quad$ with $\beta_k = \frac{(A_{k+2}-A_{k+1})(A_{k+1}(1-q)-A_k-1)}{A_{k+2}(2qA_{k+1}+1)-qA_{k+1}^2}$
7: **end for**

Output: Approximate solution x_N.

Proposition B.4. The two sequences $\{x_k\}_k$ and $\{y_k\}_k$ generated by Algorithm 14 are equal to those generated by Algorithm 28.

Proof. Without loss of generality, we can consider that a third sequence z_k is present in Algorithm 28 (although it is not computed).

Obviously, we have $x_0 = z_0 = y_0$ in both methods. Let us assume that the sequences match up to iteration k, that is, up to y_k, x_k, and z_k. Clearly, x_{k+1} is computed in the

same way in both methods as a gradient step from y_k, and it remains to compare the update rules for y_{k+1}. In Algorithm 14, we have

$$\begin{aligned} y_{k+1} =& x_k + (\tau_k - \tau_{k+1}(\tau_k - 1)(1 - q\delta_k))\,(z_k - x_k) \\ & - \frac{(\delta_k - 1)\tau_{k+1} + 1}{L}\nabla f(y_k), \end{aligned}$$

whereas in Algorithm 14, we have

$$y_{k+1} = x_k + (\beta_k + 1)\tau_k(z_k - x_k) - \frac{1 + \beta_k}{L}\nabla f(y_k).$$

By noting that $\beta_k = \tau_{k+1}(\delta_k - 1)$, we see that the coefficients in front of $\nabla f(y_k)$ match in both expressions. It remains to check that

$$(\beta_k + 1)\tau_k - (\tau_k - \tau_{k+1}(\tau_k - 1)(1 - q\delta_k))$$

is identically 0 to reach the desired statement. By substituting $\beta_k = \tau_{k+1}(\delta_k - 1)$, this expression reduces to

$$\tau_{k+1}(\delta_k(\tau_k(1 - q) + q) - 1),$$

and we have to verify that $(\delta_k(\tau_k(1 - q) + q) - 1)$ is zero. Substituting and reworking this expression using the expressions for τ_k, and δ_k, we arrive at

$$\frac{\tau_k\left((A_{k+1} - A_k)^2 - A_{k+1} - qA_{k+1}^2\right)}{(A_{k+1} - A_k)(1 + qA_{k+1})} = 0,$$

as we recognize that $(A_{k+1} - A_k)^2 - A_{k+1} - qA_{k+1}^2 = 0$ (which is the expression we used to select A_{k+1}). ■

Algorithm 29 Nesterov's method, form III

Input: L-smooth μ-strongly convex function f and initial point x_0.

1: **Initialize** $z_0 = x_0$ and $A_0 = 0$; $q = \mu/L$.
2: **for** $k = 0, \ldots$ **do**
3: $\quad A_{k+1} = \frac{2A_k + 1 + \sqrt{4A_k + 4qA_k^2 + 1}}{2(1-q)}$
4: $\quad$ set $\tau_k = \frac{(A_{k+1} - A_k)(1 + qA_k)}{A_{k+1} + 2qA_kA_{k+1} - qA_k^2}$ and $\delta_k = \frac{A_{k+1} - A_k}{1 + qA_{k+1}}$
5: $\quad y_k = x_k + \tau_k(z_k - x_k)$
6: $\quad z_{k+1} = (1 - q\delta_k)z_k + q\delta_k y_k - \frac{\delta_k}{L}\nabla f(y_k)$
7: $\quad x_{k+1} = \frac{A_k}{A_{k+1}}x_k + (1 - \frac{A_k}{A_{k+1}})z_{k+1}$
8: **end for**

Output: Approximate solution x_N.

Proposition B.5. The three sequences $\{z_k\}_k$, $\{x_k\}_k$, and $\{y_k\}_k$ generated by Algorithm 14 are equal to those generated by Algorithm 29.

Proof. Clearly, we have $x_0 = z_0 = y_0$ in both methods. Let us assume that the sequences match up to iteration k, that is, up to y_{k-1}, x_k, and z_k. Since y_k and z_{k+1} are clearly computed in the same way in both methods, we only have to verify that the update rules for x_{k+1} match. In other words, we have to verify that

$$\frac{A_k}{A_{k+1}} x_k + (1 - \frac{A_k}{A_{k+1}}) z_{k+1} = y_k - \frac{1}{L} \nabla f(y_k),$$

which, using the update rules for z_{k+1} and y_k, amounts to verifying that

$$-\frac{(A_{k+1} - A_k)^2 - A_{k+1} - q A_{k+1}^2}{L A_{k+1}(1 + q A_{k+1})} \nabla f(y_k) = 0.$$

This statement is true since we recognize $(A_{k+1} - A_k)^2 - A_{k+1} - q A_{k+1}^2 = 0$ as the expression used to select A_{k+1}. ■

B.2 Conjugate Gradient Method

Historically, Nesterov's accelerated gradient method (Nesterov, 1983) was preceded by a few other methods with optimal worst-case convergence rates $O(N^{-2})$ for smooth convex minimization. However, the alternate schemes required the capability to optimize exactly over a few dimensions—plane-searches were used in (Nemirovsky and Yudin, 1983c; Nemirovsky and Yudin, 1983b) and line-searches were used in (Nemirovsky, 1982); unfortunately these references are not available in English, and we refer to (Narkiss and Zibulevsky, 2005) for related discussions.

In this vein, accelerated methods can be obtained through their links with conjugate gradients (Algorithm 30), as a by-product of the worst-case analysis. In this section, we illustrate the absolute perfection of the connection between the OGM and conjugate gradients is absolutely perfect: an identical proof (achieving the lower bound) is valid for both methods. The conjugate gradient (CG) method for solving quadratic optimization

Algorithm 30 Conjugate gradient method

Input: L-smooth convex function f, initial point y_0, and budget N.

1: **for** $k = 0, \ldots, N-1$ **do**

2: $\quad y_{k+1} = \operatorname{argmin}_x \{ f(x) : x \in y_0 + \operatorname{span}\{\nabla f(y_0), \ldots, \nabla f(y_k)\} \}$

3: **end for**

Output: Approximate solution y_N.

problems is known to have an efficient form that does not require span-searches (which are in general too expensive to be of any practical interest); see, for example, (Nocedal and Wright, 2006). Beyond quadratics, it is generally not possible to reformulate the CG method in an efficient way. However, it is possible to find other methods for which the same worst-case analysis applies, and it turns out that the OGM is one of them—see (Drori and Taylor, 2020) for details. Similarly, by slightly weakening the analysis of

the CG method, one can find other methods, such as Nesterov's accelerated gradient (see Remark B.1 below for more details).

More precisely, recall the previous definition for the sequence $\{\theta_{k,N}\}_k$, defined in (4.8):

$$\theta_{k+1,N} = \begin{cases} \frac{1+\sqrt{4\theta_{k,N}^2+1}}{2} & \text{if } k \leq N-2 \\ \frac{1+\sqrt{8\theta_{k,N}^2+1}}{2} & \text{if } k = N-1. \end{cases}$$

As a result of the worst-case analysis presented below, all methods satisfying

$$\langle \nabla f(y_i); y_i - \left[\left(1 - \frac{1}{\theta_{i,N}}\right) \left(y_{i-1} - \tfrac{1}{L}\nabla f(y_{i-1})\right) + \frac{1}{\theta_{i,N}} \left(y_0 - \tfrac{2}{L} \sum_{j=0}^{i-1} \theta_{j,N} \nabla f(y_j) \right) \right] \rangle \leq 0 \tag{B.1}$$

achieve the optimal worst-case complexity of smooth convex minimization that is provided by Theorem 4.7. On the one hand, the CG ensures that this inequality holds thanks to its span-searches (which ensure the orthogonality of successive search directions); that is,

$$\begin{aligned} \langle \nabla f(y_i); y_i - y_{i-1} + \frac{1}{\theta_{i,N}}(y_{i-1} - y_0) \rangle &= 0 \\ \langle \nabla f(y_i); \nabla f(y_0) \rangle &= 0 \\ &\vdots \\ \langle \nabla f(y_i); \nabla f(y_{i-1}) \rangle &= 0. \end{aligned}$$

On the other hand, the OGM enforces this inequality by using

$$y_i = \left(1 - \frac{1}{\theta_{i,N}}\right) \left(y_{i-1} - \tfrac{1}{L}\nabla f(y_{i-1})\right) + \frac{1}{\theta_{i,N}} \left(y_0 - \tfrac{2}{L} \sum_{j=0}^{i-1} \theta_{j,N} \nabla f(y_j) \right).$$

Optimized and Conjugate Gradient Methods: Worst-case Analyses

The worst-case analysis below relies on the same potentials used for the optimized gradient method; see Theorem 4.4 and Lemma 4.5.

Theorem B.1. Let f be an L-smooth convex function, $N \in \mathbb{N}$ and some $x_\star \in \operatorname{argmin}_x f(x)$. The iterates of the conjugate gradient method (CG, Algorithm 30) and of all methods whose iterates are compliant with (B.1) satisfy

$$f(y_N) - f(x_\star) \leq \frac{L\|y_0 - x_\star\|_2^2}{2\theta_{N,N}^2},$$

for all $y_0 \in \mathbb{R}^d$.

Proof. The result is obtained from the same potential as that used for the OGM, obtained from further inequalities. That is, we first perform a weighted sum of the following inequalities.

- Smoothness and convexity of f between y_{k-1} and y_k with weight $\lambda_1 = 2\theta_{k-1,N}^2$:

$$\begin{aligned} 0 \geq & f(y_k) - f(y_{k-1}) + \langle \nabla f(y_k); y_{k-1} - y_k \rangle \\ & + \frac{1}{2L} \|\nabla f(y_k) - \nabla f(y_{k-1})\|_2^2. \end{aligned}$$

- Smoothness and convexity of f between $x_\star$ and y_k with weight $\lambda_2 = 2\theta_{k,N}$:

$$0 \geq f(y_k) - f(x_\star) + \langle \nabla f(y_k); x_\star - y_k \rangle + \frac{1}{2L} \|\nabla f(y_k)\|_2^2.$$

- Search procedure to obtain y_k, with weight $\lambda_3 = 2\theta_{k,N}^2$:

$$0 \geq \langle \nabla f(y_k); y_k - \left[\left(1 - \frac{1}{\theta_{k,N}} \right) \left(y_{k-1} - \tfrac{1}{L} \nabla f(y_{k-1}) \right) + \frac{1}{\theta_{k,N}} z_k \right] \rangle,$$

  where we used $z_k := y_0 - \frac{2}{L} \sum_{j=0}^{k-1} \theta_{j,N} \nabla f(y_j)$.

The weighted sum is a valid inequality:

$$\begin{aligned} 0 \geq & \lambda_1 [f(y_k) - f(y_{k-1}) + \langle \nabla f(y_k); y_{k-1} - y_k \rangle \\ & \quad + \frac{1}{2L} \|\nabla f(y_k) - \nabla f(y_{k-1})\|_2^2] \\ & + \lambda_2 [f(y_k) - f(x_\star) + \langle \nabla f(y_k); x_\star - y_k \rangle + \frac{1}{2L} \|\nabla f(y_k)\|_2^2] \\ & + \lambda_3 [\langle \nabla f(y_k); y_k - \left[\left(1 - \frac{1}{\theta_{k,N}} \right) \left(y_{k-1} - \tfrac{1}{L} \nabla f(y_{k-1}) \right) \right. \\ & \qquad \left. + \frac{1}{\theta_{k,N}} z_k \right] \rangle]. \end{aligned}$$

Substituting z_{k+1}, the previous inequality can be reformulated exactly as

$$\begin{aligned} 0 \geq & 2\theta_{k,N}^2 \left(f(y_k) - f_\star - \frac{1}{2L} \|\nabla f(y_k)\|_2^2 \right) + \frac{L}{2} \|z_{k+1} - x_\star\|_2^2 \\ & - 2\theta_{k-1,N}^2 \left(f(y_{k-1}) - f_\star - \frac{1}{2L} \|\nabla f(y_{k-1})\|_2^2 \right) - \frac{L}{2} \|z_k - x_\star\|_2^2 \\ & + 2 \left(\theta_{k-1,N}^2 - \theta_{k,N}^2 + \theta_{k,N} \right) \left(f(y_k) - f_\star + \frac{1}{2L} \|\nabla f(y_k)\|_2^2 \right) \\ & + 2 \left(\theta_{k-1,N}^2 - \theta_{k,N}^2 + \theta_{k,N} \right) \langle \nabla f(y_k); y_{k-1} - \tfrac{1}{L} \nabla f(y_{k-1}) - y_k \rangle. \end{aligned}$$

We reach the desired inequality by selecting $\theta_{k,N}$ that satisfies $\theta_{k,N} \geq \theta_{k-1,N}$ and

$$\theta_{k-1,N}^2 - \theta_{k,N}^2 + \theta_{k,N} = 0,$$

thereby reaching the same potential as in Theorem 4.4.

To obtain the technical lemma that allows us to bound the final $f(y_N) - f_\star$, we follow the same steps with the following inequalities.

- Smoothness and convexity of f between y_{k-1} and y_k with weight $\lambda_1 = 2\theta_{N-1,N}^2$:
$$0 \geq f(y_N) - f(y_{N-1}) + \langle \nabla f(y_N); y_{N-1} - y_N \rangle + \frac{1}{2L}\|\nabla f(y_N) - \nabla f(y_{N-1})\|_2^2.$$
- Smoothness and convexity of f between $x_\star$ and y_k with weight $\lambda_2 = \theta_{N,N}$:
$$0 \geq f(y_N) - f(x_\star) + \langle \nabla f(y_N); x_\star - y_N \rangle + \frac{1}{2L}\|\nabla f(y_N)\|_2^2.$$
- Search procedure to obtain y_N, with weight $\lambda_3 = \theta_{N,N}^2$:
$$0 \geq \langle \nabla f(y_N); y_N - \left[\left(1 - \frac{1}{\theta_{N,N}}\right) \left(y_{N-1} - \tfrac{1}{L}\nabla f(y_{N-1})\right) + \frac{1}{\theta_{N,N}} z_N \right] \rangle.$$

The weighted sum can then be reformulated as:
$$\begin{aligned} 0 \geq & \theta_{N,N}^2 \left(f(y_N) - f_\star\right) + \frac{L}{2}\|z_N - \tfrac{\theta_{N,N}}{L}\nabla f(y_N) - x_\star\|_2^2 \\ & - 2\theta_{N-1,N}^2 \left(f(y_{N-1}) - f_\star - \frac{1}{2L}\|\nabla f(y_{N-1})\|_2^2\right) - \frac{L}{2}\|z_N - x_\star\|_2^2 \\ & + \left(2\theta_{N-1,N}^2 - \theta_{N,N}^2 + \theta_{N,N}\right)\left(f(y_N) - f_\star + \frac{1}{2L}\|\nabla f(y_N)\|_2^2\right) \\ & + \left(2\theta_{N-1,N}^2 - \theta_{N,N}^2 + \theta_{N,N}\right)\langle \nabla f(y_N); y_{N-1} - \tfrac{1}{L}\nabla f(y_{N-1}) - y_N \rangle, \end{aligned}$$
thus reaching the desired inequality, as in Lemma 4.5, by selecting $\theta_{N,N}$ that satisfies $\theta_{N,N} \geq \theta_{N-1,N}$ and
$$2\theta_{N-1,N}^2 - \theta_{N,N}^2 + \theta_{N,N}.$$
Hence, the potential argument from Corollary 4.6 applies as such, and we reach the desired conclusion. In other words, for all $k \in \{0, \dots, N\}$, one can define
$$\phi_k \triangleq 2\theta_{k-1,N}^2 \left(f(y_{k-1}) - f_\star - \frac{1}{2L}\|\nabla f(y_{k-1})\|_2^2\right) + \frac{L}{2}\|z_k - x_\star\|_2^2$$
and
$$\phi_{N+1} \triangleq \theta_{N,N}^2 \left(f(y_N) - f_\star\right) + \frac{L}{2}\|z_N - \tfrac{\theta_{N,N}}{L}\nabla f(y_N) - x_\star\|_2^2$$
and reach the desired statement by chaining the inequalities:
$$\theta_{N,N}^2 (f(y_N) - f_\star) \leq \phi_{N+1} \leq \phi_N \leq \ldots \leq \phi_0 = \frac{L}{2}\|y_0 - x_\star\|_2^2. \quad \blacksquare$$

Remark B.1. It is possible to further exploit the conjugate gradient method to design practical accelerated methods in different settings, such as that of Nesterov (1983). This point of view has been exploited in (Narkiss and Zibulevsky, 2005; Karimi and Vavasis, 2016; Karimi and Vavasis, 2017; Diakonikolas and Orecchia, 2019a), among others. The link between the CG method and the OGM presented in this section is due to Drori and Taylor (2020), though with a different presentation that does not involve the potential function.

B.3 Acceleration Without Monotone Backtracking

B.3.1 FISTA without Monotone Backtracking

In this section, we show how to incorporate backtracking strategies that may not satisfy $L_{k+1} \geq L_k$, which is important in practice. The developments are essentially the same; one possible trick is to incorporate all the knowledge about L_k in A_k. That is, we use a rescaled shape for the potential function:

$$\phi_k \triangleq B_k(f(x_k) - f_\star) + \frac{1 + \mu B_k}{2}\|z_k - x_\star\|_2^2,$$

where without the backtracking strategy, $B_k = \frac{A_k}{L}$. This seemingly cosmetic change allows ϕ_k to depend on L_k solely via B_k, and it applies to both backtracking methods presented in Section 4 (Section 4.7).

The idea used to obtain both methods below is that one can perform the same computations as in Algorithm 14, replacing A_k by $L_{k+1}B_k$ and A_{k+1} by $L_{k+1}A_{k+1}$ at iteration k. Thus, as in previous versions, only the current approximate Lipschitz constant L_{k+1} is used at iteration k: previous approximations were only used to compute B_k.

Algorithm 31 Strongly convex FISTA (general initialization of L_{k+1})

Input: An L-smooth (possibly μ-strongly) convex function f, a convex function h with proximal operator available, an initial point x_0, and an initial estimate $L_0 > \mu$.

Initialize $z_0 = x_0$, $B_0 = 0$, and some $\alpha > 1$.
for $k = 0, \ldots$ **do**
  Pick $L_{k+1} \in [L_0, L_k]$.
  loop
    set $q_{k+1} = \mu / L_{k+1}$,
    $B_{k+1} = \frac{2L_{k+1}B_k + 1 + \sqrt{4L_{k+1}B_k + 4\mu L_{k+1}B_k^2 + 1}}{2(L_{k+1} - \mu)}$
    set $\tau_k = \frac{(B_{k+1} - B_k)(1 + \mu B_k)}{(B_{k+1} + 2\mu B_k B_{k+1} - \mu B_k^2)}$ and $\delta_k = L_{k+1}\frac{B_{k+1} - B_k}{1 + \mu B_{k+1}}$
    $y_k = x_k + \tau_k(z_k - x_k)$
    $x_{k+1} = \mathrm{prox}_{h/L_{k+1}}\left(y_k - \frac{1}{L_{k+1}}\nabla f(y_k)\right)$
    $z_{k+1} = (1 - q_{k+1}\delta_k)z_k + q_{k+1}\delta_k y_k + \delta_k\,(x_{k+1} - y_k)$
    if (4.21) holds **then**
      break {Iterates accepted; k will be incremented.}
    else
      $L_{k+1} = \alpha L_{k+1}$ {Iterates not accepted; compute new L_{k+1}.}
    end if
  end loop
end for

Output: Approximate solution x_{k+1}.

The proof follows the same lines as used for FISTA (Algorithm 4.20). In this case, f is assumed to be smooth and convex over $\mathbb{R}^d$ (i.e., it has full domain, $\mathbf{dom}\, f = \mathbb{R}^d$), and we are therefore allowed to evaluate gradients of f outside of the domain of h.

Theorem B.2. Let $f \in \mathcal{F}_{\mu,L}$ (with full domain, $\mathbf{dom}\, f = \mathbb{R}^d$), h be a closed convex proper function, $x_\star \in \operatorname{argmin}_x \{F(x) \triangleq f(x) + h(x)\}$, and $k \in \mathbb{N}$. For any $x_k, z_k \in \mathbb{R}^d$ and $B_k \geq 0$, the iterates of Algorithm 31 that satisfy (4.21) also satisfy

$$\begin{aligned} B_{k+1}(F(x_{k+1}) - F_\star) + \frac{1+\mu B_{k+1}}{2}\|z_{k+1} - x_\star\|_2^2 \\ \leq B_k(F(x_k) - F_\star) + \frac{1+\mu B_k}{2}\|z_k - x_\star\|_2^2, \end{aligned}$$

with $B_{k+1} = \frac{2L_{k+1}B_k + 1 + \sqrt{4L_{k+1}B_k + 4\mu L_{k+1}B_k^2 + 1}}{2(L_{k+1} - \mu)}$.

Proof. The proof consists of a weighted sum of the following inequalities.

- Strong convexity of f between $x_\star$ and y_k with weight $\lambda_1 = B_{k+1} - B_k$:
$$f_\star \geq f(y_k) + \langle \nabla f(y_k); x_\star - y_k\rangle + \frac{\mu}{2}\|x_\star - y_k\|_2^2.$$
- Strong convexity of f between x_k and y_k with weight $\lambda_2 = B_k$:
$$f(x_k) \geq f(y_k) + \langle \nabla f(y_k); x_k - y_k\rangle.$$
- Smoothness of f between y_k and x_{k+1} (*descent lemma*) with weight $\lambda_3 = B_{k+1}$:
$$f(y_k) + \langle \nabla f(y_k); x_{k+1} - y_k\rangle + \frac{L_{k+1}}{2}\|x_{k+1} - y_k\|_2^2 \geq f(x_{k+1}).$$
- Convexity of h between $x_\star$ and x_{k+1} with weight $\lambda_4 = B_{k+1} - B_k$:
$$h(x_\star) \geq h(x_{k+1}) + \langle g_h(x_{k+1}); x_\star - x_{k+1}\rangle,$$
with $g_h(x_{k+1}) \in \partial h(x_{k+1})$ and $x_{k+1} = y_k - \frac{1}{L_{k+1}}(\nabla f(y_k) + g_h\ (x_{k+1}))$.
- Convexity of h between x_k and x_{k+1} with weight $\lambda_5 = B_k$:
$$h(x_k) \geq h(x_{k+1}) + \langle g_h(x_{k+1}); x_k - x_{k+1}\rangle.$$

We obtain the following inequality:

$$\begin{aligned} 0 \geq & \lambda_1[f(y_k) - f_\star + \langle \nabla f(y_k); x_\star - y_k\rangle + \frac{\mu}{2}\|x_\star - y_k\|_2^2] \\ & + \lambda_2[f(y_k) - f(x_k) + \langle \nabla f(y_k); x_k - y_k\rangle] \\ & + \lambda_3[f(x_{k+1}) - (f(y_k) + \langle \nabla f(y_k); x_{k+1} - y_k\rangle \\ & \quad + \frac{L_{k+1}}{2}\|x_{k+1} - y_k\|_2^2)] \\ & + \lambda_4[h(x_{k+1}) - h(x_\star) + \langle g_h(x_{k+1}); x_\star - x_{k+1}\rangle] \\ & + \lambda_5[h(x_{k+1}) - h(x_k) + \langle g_h(x_{k+1}); x_k - x_{k+1}\rangle]. \end{aligned}$$

Substituting the y_k, x_{k+1}, and z_{k+1} with

$$\begin{aligned} y_k &= x_k + \tau_k(z_k - x_k) \\ x_{k+1} &= y_k - \frac{1}{L_{k+1}}(\nabla f(y_k) + g_h(x_{k+1})) \\ z_{k+1} &= (1 - q_{k+1}\delta_k)z_k + q_{k+1}\delta_k y_k + \delta_k\left(x_{k+1} - y_k\right), \end{aligned}$$

after some basic but tedious algebra, yields

$$\begin{aligned} &B_{k+1}(f(x_{k+1}) + h(x_{k+1}) - f(x_\star) - h(x_\star)) + \frac{1 + B_{k+1}\mu}{2}\|z_{k+1} - x_\star\|_2^2 \\ \leq &B_k(f(x_k) + h(x_k) - f(x_\star) - h(x_\star)) + \frac{1 + B_k\mu}{2}\|z_k - x_\star\|_2^2 \\ &+ \frac{L_{k+1}(B_k - B_{k+1})^2 - B_{k+1} - \mu B_{k+1}^2}{1 + \mu B_{k+1}} \\ &\quad \times \frac{1}{2L_{k+1}}\|\nabla f(y_k) + g_h(x_{k+1})\|_2^2 \\ &- \frac{B_k^2(B_{k+1} - B_k)(1 + \mu B_k)(1 + \mu B_{k+1})}{\left(B_{k+1} + 2\mu B_k B_{k+1} - \mu B_k^2\right)^2}\frac{\mu}{2}\|x_k - z_k\|_2^2. \end{aligned}$$

Then, choosing B_{k+1} such that $B_{k+1} \geq B_k$ and

$$L_{k+1}(B_k - B_{k+1})^2 - B_{k+1} - \mu B_{k+1}^2 = 0,$$

yields the desired result:

$$\begin{aligned} &B_{k+1}(f(x_{k+1}) + h(x_{k+1}) - f(x_\star) - h(x_\star)) + \frac{1 + B_{k+1}\mu}{2}\|z_{k+1} - x_\star\|_2^2 \\ \leq &B_k(f(x_k) + h(x_k) - f(x_\star) - h(x_\star)) + \frac{1 + B_k\mu}{2}\|z_k - x_\star\|_2^2. \quad \blacksquare \end{aligned}$$

Finally, we obtain a complexity guarantee by adapting the potential argument (4.5) and by noting that B_{k+1} is a decreasing function of L_{k+1} (whose maximal value is αL, assuming $L_0 < L$; otherwise, its maximal value is L_0). The growth rate of B_k in the smooth convex setting remains unchanged (see (4.14)) since we have

$$B_{k+1} \geq \frac{\left(\frac{1}{2} + \sqrt{B_k L_{k+1}}\right)^2}{L_{k+1}},$$

and hence, $\sqrt{B_{k+1}} \geq \frac{1}{2\sqrt{L_{k+1}}} + \sqrt{B_k}$. Therefore, $B_k \geq \left(\frac{k}{2\sqrt{\ell}}\right)^2$ with $\ell = \max\{L_0, \alpha L\}$ and $L_{k+1} \leq \ell$. As for the geometric rate, we similarly obtain

$$B_{k+1} \geq B_k \frac{\left(1 + \sqrt{\frac{\mu}{L_{k+1}}}\right)}{1 - \frac{\mu}{L_{k+1}}} = \frac{B_k}{1 - \sqrt{\frac{\mu}{L_{k+1}}}},$$

and therefore, $B_{k+1} \geq (1 - \sqrt{\frac{\mu}{\ell}})^{-1} B_k$.

Corollary B.3. Let $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ (with full domain, $\mathbf{dom}\, f = \mathbb{R}^d$), h be a closed convex proper function and $x_\star \in \operatorname{argmin}_x \{F(x) \triangleq f(x) + h(x)\}$. For any $N \in \mathbb{N}$, $N \geq 1$, and $x_0 \in \mathbb{R}^d$, the output of Algorithm 31 satisfies

$$F(x_N) - F_\star \leq \min\left\{\frac{2}{N^2}, \left(1 - \sqrt{\frac{\mu}{\ell}}\right)^N\right\} \ell \|x_0 - x_\star\|_2^2,$$

with $\ell = \max\{\alpha L, L_0\}$.

Proof. We assume that $L > L_0$ since otherwise, $f \in \mathcal{F}_{\mu,L_0}$ and the proof directly follows from the case without backtracking. The chained potential argument (4.5) can be used as before. Using $B_0 = 0$, we reach

$$F(x_N) - F_\star \leq \frac{\|x_0 - x_\star\|_2^2}{2B_N}.$$

Our previous bounds on B_N yields the desired result, using

$$B_1 = \frac{1}{L_{k+1} - \mu} \geq \frac{2\ell^{-1}}{1 - \frac{\mu}{\ell}} = \frac{2\ell^{-1}}{\left(1 - \sqrt{\frac{\mu}{\ell}}\right)\left(1 + \sqrt{\frac{\mu}{\ell}}\right)} \geq \frac{\ell^{-1}}{1 - \sqrt{\frac{\mu}{\ell}}},$$

and hence, $B_N \geq \ell^{-1}\left(1 - \sqrt{\frac{\mu}{\ell}}\right)^{-N}$ as well as $B_k \geq \left(\frac{k}{2\sqrt{\ell}}\right)^2$. ∎

B.3.2 Another Accelerated Method without Monotone Backtracking

Just as for FISTA, we can perform the same cosmetic change to Algorithm 20 for incorporating a non-monotonic estimations of the Lipschitz constant. The proof is therefore essentially that of Algorithm 20.

Theorem B.4. Let $h \in \mathcal{F}_{0,\infty}$, $f \in \mathcal{F}_{\mu,L}(\mathbf{dom}\, h)$, $x_\star \in \operatorname{argmin}_x \{F(x) \triangleq f(x) + h(x)\}$, and $k \in \mathbb{N}$. For any $x_k, z_k \in \mathbb{R}^d$ and $B_k \geq 0$, the iterates of Algorithm 32 that satisfy (4.21) also satisfy

$$\begin{aligned} B_{k+1}(F(x_{k+1}) - F_\star) + \frac{1 + \mu B_{k+1}}{2}\|z_{k+1} - x_\star\|_2^2 \\ \leq B_k(F(x_k) - F_\star) + \frac{1 + \mu B_k}{2}\|z_k - x_\star\|_2^2, \end{aligned}$$

with $B_{k+1} = \frac{2L_{k+1}B_k + 1 + \sqrt{4L_{k+1}B_k + 4\mu L_{k+1}B_k^2 + 1}}{2(L_{k+1} - \mu)}$.

Proof. First, $\{z_k\}_k$ is in $\mathbf{dom}\, h$ by construction—it is the output of a proximal/projection step. Furthermore, we have $0 \leq \frac{B_k}{B_{k+1}} \leq 1$ given that $B_{k+1} \geq B_k \geq 0$. A direct consequence is that since $z_0 = x_0 \in \mathbf{dom}\, h$, all subsequent $\{y_k\}_k$ and $\{x_k\}_k$ are also in $\mathbf{dom}\, h$ (as they are obtained from convex combinations of feasible points).

The rest of the proof consists of a weighted sum of the following inequalities (which are valid due to the feasibility of the iterates).

Algorithm 32 A proximal accelerated gradient (general initialization of L_{k+1})

Input: $h \in \mathcal{F}_{0,\infty}$ with proximal operator available, $f \in \mathcal{F}_{\mu,L}(\mathbf{dom}\, h)$, an initial point $x_0 \in \mathbf{dom}\, h$, and an initial estimate $L_0 > \mu$.

Initialize $z_0 = x_0$, $A_0 = 0$, and some $\alpha > 1$.
for $k = 0, \ldots$ **do**
  Pick $L_{k+1} \in [L_0, L_k]$.
  loop
    Set $q_{k+1} = \mu / L_{k+1}$,
    $B_{k+1} = \frac{2L_{k+1}B_k + 1 + \sqrt{4L_{k+1}B_k + 4\mu L_{k+1}B_k^2 + 1}}{2(L_{k+1} - \mu)}$
    Set $\tau_k = \frac{L_{k+1}(B_{k+1} - B_k)(1 + \mu B_k)}{L_{k+1}(B_{k+1} + 2\mu B_k B_{k+1} - \mu B_k^2)}$ and $\delta_k = L_{k+1}\frac{B_{k+1} - B_k}{1 + \mu B_{k+1}}$
    $y_k = x_k + \tau_k(z_k - x_k)$
    $z_{k+1} = \mathrm{prox}_{\delta_k h / L_{k+1}}\left((1 - q_{k+1}\delta_k)z_k + q_{k+1}\delta_k y_k - \frac{\delta_k}{L_{k+1}}\nabla\, f(y_k)\right)$
    $x_{k+1} = \frac{A_k}{A_{k+1}}x_k + (1 - \frac{A_k}{A_{k+1}})z_{k+1}$
    if (4.21) holds **then**
      break {Iterates accepted; k will be incremented.}
    else
      $L_{k+1} = \alpha L_{k+1}$ {Iterates not accepted; compute new L_{k+1}.}
    end if
  end loop
end for

Output: An approximate solution x_{k+1}.

- Strong convexity of f between $x_\star$ and y_k with weight $\lambda_1 = B_{k+1} - B_k$:
$$f(x_\star) \geq f(y_k) + \langle \nabla f(y_k); x_\star - y_k \rangle + \frac{\mu}{2}\|x_\star - y_k\|_2^2.$$
- Convexity of f between x_k and y_k with weight $\lambda_2 = B_k$:
$$f(x_k) \geq f(y_k) + \langle \nabla f(y_k); x_k - y_k \rangle.$$
- Smoothness of f between y_k and x_{k+1} (*descent lemma*) with weight $\lambda_3 = B_{k+1}$:
$$f(y_k) + \langle \nabla f(y_k); x_{k+1} - y_k \rangle + \frac{L_{k+1}}{2}\|x_{k+1} - y_k\|_2^2 \geq f(x_{k+1}).$$
- Convexity of h between $x_\star$ and z_{k+1} with weight $\lambda_4 = B_{k+1} - B_k$:
$$h(x_\star) \geq h(z_{k+1}) + \langle g_h(z_{k+1}); x_\star - z_{k+1} \rangle,$$
with $g_h(z_{k+1}) \in \partial h(z_{k+1})$ and $z_{k+1} = (1 - q\delta_k)z_k + q\delta_k y_k - \frac{\delta_k}{L_{k+1}}(\nabla f(y_k) + g_h(z_{k+1}))$.
- Convexity of h between x_k and x_{k+1} with weight $\lambda_5 = B_k$:
$$h(x_k) \geq h(x_{k+1}) + \langle g_h(x_{k+1}); x_k - x_{k+1} \rangle,$$
with $g_h(x_{k+1}) \in \partial h(x_{k+1})$.

- Convexity of h between z_{k+1} and x_{k+1} with weight $\lambda_6 = B_{k+1} - B_k$:

$$h(z_{k+1}) \geq h(x_{k+1}) + \langle g_h(x_{k+1}); z_{k+1} - x_{k+1} \rangle.$$

We obtain the following inequality:

$$\begin{aligned}
0 \geq & \lambda_1[f(y_k) - f_\star + \langle \nabla f(y_k); x_\star - y_k \rangle + \frac{\mu}{2}\|x_\star - y_k\|_2^2] \\
& + \lambda_2[f(y_k) - f(x_k) + \langle \nabla f(y_k); x_k - y_k \rangle] \\
& + \lambda_3[f(x_{k+1}) - (f(y_k) + \langle \nabla f(y_k); x_{k+1} - y_k \rangle \\
& \quad + \frac{L_{k+1}}{2}\|x_{k+1} - y_k\|_2^2)] \\
& + \lambda_4[h(z_{k+1}) - h(x_\star) + \langle g_h(z_{k+1}); x_\star - z_{k+1} \rangle] \\
& + \lambda_5[h(x_{k+1}) - h(x_k) + \langle g_h(x_{k+1}); x_k - x_{k+1} \rangle] \\
& + \lambda_6[h(x_{k+1}) - h(z_{k+1}) + \langle g_h(x_{k+1}); z_{k+1} - x_{k+1} \rangle].
\end{aligned}$$

Substituting the y_k, z_{k+1}, and x_{k+1} by

$$\begin{aligned}
y_k &= x_k + \tau_k(z_k - x_k) \\
z_{k+1} &= (1 - q_{k+1}\delta_k)z_k + q_{k+1}\delta_k y_k - \frac{\delta_k}{L_{k+1}}(\nabla f(y_k) + g_h(z_{k+1})) \\
x_{k+1} &= \frac{B_k}{B_{k+1}}x_k + \left(1 - \frac{B_k}{B_{k+1}}\right) z_{k+1},
\end{aligned}$$

and algebra allows us to obtain the following reformulation:

$$\begin{aligned}
& B_{k+1}(f(x_{k+1}) + h(x_{k+1}) - f(x_\star) - h(x_\star)) + \frac{1 + \mu B_{k+1}}{2}\|z_{k+1} - x_\star\|_2^2 \\
\leq & B_k(f(x_k) + h(x_k) - f(x_\star) - h(x_\star)) + \frac{1 + \mu B_k}{2}\|z_k - x_\star\|_2^2 \\
& + \frac{(B_k - B_{k+1})^2 \left(L_{k+1}(B_k - B_{k+1})^2 - B_{k+1} - \mu B_{k+1}^2\right)}{B_{k+1}(1 + \mu B_{k+1})^2} \\
& \qquad \times \frac{1}{2}\|\nabla f(y_k) + g_h(z_{k+1})\|_2^2 \\
& - \frac{B_k^2(B_{k+1} - B_k)(1 + \mu B_k)(1 + \mu B_{k+1})}{\left(B_{k+1} + 2\mu B_k B_{k+1} - \mu B_k^2\right)^2} \frac{\mu}{2}\|x_k - z_k\|_2^2.
\end{aligned}$$

The desired inequality follows from selecting B_{k+1} such that $B_{k+1} \geq B_k$ and

$$L_{k+1}(B_k - B_{k+1})^2 - B_{k+1} - \mu B_{k+1}^2 = 0,$$

thereby yielding

$$\begin{aligned}
& B_{k+1}(f(x_{k+1}) + h(x_{k+1}) - f(x_\star) - h(x_\star)) + \frac{1 + \mu B_{k+1}}{2}\|z_{k+1} - x_\star\|_2^2 \\
\leq & B_k(f(x_k) + h(x_k) - f(x_\star) - h(x_\star)) + \frac{1 + B_k\mu}{2}\|z_k - x_\star\|_2^2. \quad \blacksquare
\end{aligned}$$

The final corollary follows from the same arguments as those used for Corollary B.3. It provides the final bound for Algorithm 32.

Corollary B.5. Let $h \in \mathcal{F}_{0,\infty}$, $f \in \mathcal{F}_{\mu,L}(\mathbf{dom}\, h)$, and $x_\star \in \operatorname{argmin}_x \{F(x) \triangleq f(x)+h(x)\}$. For any $N \in \mathbb{N}$, $N \geq 1$, and $x_0 \in \mathbb{R}^d$, the output of Algorithm 32 satisfies

$$F(x_N) - F_\star \leq \min\left\{\frac{2}{N^2}, \left(1-\sqrt{\frac{\mu}{\ell}}\right)^{-N}\right\} \ell \|x_0 - x_\star\|_2^2,$$

with $\ell = \max\{\alpha L, L_0\}$.

Proof. The proof follows the same arguments as those for Corollary B.3, using the potential from Theorem B.4 and the fact that the output of the algorithm satisfies (4.21). ■

C

On Worst-case Analyses for First-order Methods

C.1 Principled Approaches to Worst-case Analyses

In this section, we show that obtaining convergence rates and proofs can be framed as finding feasible points to certain convex problems. More precisely, all convergence guarantees from Section 4 and Section 5 can be obtained as feasible points to certain linear matrix inequalities (LMI). As we see in what follows, this approach can be seen as a *principled* approach to worst-case analysis of first-order methods: the approach fails only when no such guarantees can be found. The purpose of this section is to provide complete examples of the LMIs for a few cases of interest: analyses of gradient and accelerated gradient methods, as well as pointers to the relevant literature. We provide a full derivation for the base case, and leave advanced ones as exercises for the reader. Notebooks for obtaining the corresponding LMIs are provided in Section C.5.

The elements of this section are largely inspired by the presentation of Taylor and Bach (2019) with elements borrowed from the presentation of Taylor, Hendrickx and Glineur (2017), which is itself largely inspired by that of Drori and Teboulle (2014). The arguments are also similar to the line of work by Lessard, Recht and Packard (2016) and follow-up works, see, e.g., (Fazlyab *et al.*, 2018; Hu and Lessard, 2017). The latter line of works is similar in spirit to the former, but framed in control-theoretic terms, via so-called *integral quadratic constraints*, popularized by Megretski and Rantzer (1997).

These techniques are analogous and mostly differs in their presentation styles. Roughly speaking, they can be seen as *dual* to each others. That is, whereas the *performance estimation* viewpoint stems from the problem of computing worst-case scenarios and approaches worst-case guarantees as feasible point to the corresponding dual problems, the *integral quadratic constraint* approach directly starts from the problem of performing linear combination of inequalities, which is exactly the dual problem to that of computing worst-case scenarios. Depending on the background of the researchers involved in a work on one of those topics, things might therefore be named in different ways. We insist on

the fact that those are really two facets of the same coin with only subtle differences in terms of presentations.

We choose to take the performance estimation viewpoint as using the definition of a "worst-case" allows to carefully select the most appropriate set of inequalities to be used. Informally, this advantageous construction allows certifying the approach to provide meaningful worst-case guarantees: either the approach provides a satisfying worst-case guarantee, or there exists a non-satisfying counterexample, invalidating the existence of any satisfying guarantee of the desired form.

Further discussions and a more thorough list of references are provided in Section C.5. Readability in mind, the presentation focuses on some examples of interest rather than on a general framework. We refer to (Drori and Teboulle, 2014; Taylor *et al.*, 2017a; Taylor *et al.*, 2017c) for more details.

C.2 Worst-case Analysis as Optimization/Feasibility Problems

In this section, we provide examples illustrating the type of problems that can be used for obtaining worst-case guarantees. The base idea underlying the technique is that worst-case scenarios are by definition solutions to certain optimization problems. In the context of first-order convex optimization methods, those worst-case scenarios correspond to solutions to linear semidefinite programs (SDP), which are convex; see, e.g., (Vandenberghe and Boyd, 1999). It nicely follows from this theory that any worst-case guarantee (i.e., any upper bound on a worst-case performance) can be formulated as a feasible point to the dual problem to that of finding worst-case scenarios. Equivalently, those dual solutions correspond to appropriate weighted sums of inequalities, whose weights correspond to the values of the dual variables. Proofs from Section 4 and Section 5 correspond to such dual certificates.

Those statements are made more precise in the next sections. We begin by providing a few examples of LMIs that can be used for designing worst-case guarantees.

Preview: worst-case guarantees via LMIs. Perhaps the most basic LMI that can be presented for obtaining worst-case guarantees concerns gradient descent and its convergence in terms of distance to an optimal point. We present it for simplicity, as the corresponding LMI only involves very few variables. This LMI has also relatively simple solutions. As our target here is to present the approach, we let finding their solutions as exercises. We present the LMIs in their most *raw* forms, even without a few direct simplifications.

Note that those LMIs always involve $n(n-1)$ "dual" variables (the precise meaning of *dual* becomes clear in the sequel), where n is the number of points at which the type of guarantee under consideration requires using or specifying a function or gradient evaluation (either in the algorithm or for computing the value of the guarantee). In the following example, we need two dual variables because the guarantee only requires using

two gradients of f, namely $\nabla f(x_k)$ (for expressing a gradient step $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$) and $\nabla f(x_\star)$ (for expressing optimality of $x_\star$ as $\nabla f(x_\star) = 0$).

Theorem C.1. Let $\tau \geq 0$ and $\gamma_k \in \mathbb{R}$. The inequality

$$\|x_{k+1} - x_\star\|_2^2 \leq \tau \|x_k - x_\star\|_2^2 \tag{C.1}$$

holds for all $d \in \mathbb{N}$, all $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, all $x_k, x_{k+1}, x_\star \in \mathbb{R}^d$ (such that $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$ and $\nabla f(x_\star) = 0$), if and only if

$$\exists \lambda_1, \lambda_2 \geq 0 : \begin{cases} \lambda_1 = \lambda_2 \\ 0 \preceq \begin{bmatrix} \tau - 1 + \frac{\mu L(\lambda_1+\lambda_2)}{2(L-\mu)} & \gamma_k - \frac{L\lambda_1+\mu\lambda_2}{2(L-\mu)} \\ \gamma_k - \frac{L\lambda_1+\mu\lambda_2}{2(L-\mu)} & -\gamma_k^2 + \frac{\lambda_1+\lambda_2}{2(L-\mu)} \end{bmatrix}. \end{cases} \tag{C.2}$$

We emphasize that the message underlying Theorem C.1 is that verifying a worst-case convergence guarantee of the form (C.1) boils down to verifying the feasibility of a certain convex problem. It is relatively straightforward to convert a feasible point of (C.2) to a proof that only consists of a weighted linear combination of inequalities, see, e.g., (Taylor *et al.*, 2018b, Theorem 3.1). The corresponding weights are the values of the multipliers (that is, in Theorem C.1, the weights are λ_1 and λ_2) as showcased in Section 4 and Section 5.

As we see in Section C.3, changing the Lyapunov, or potential, function to be verified also changes the LMI to be solved. The desired LMI can be obtained following a principled approach presented in the sequel. In particular, the following result is slightly more complicated and corresponds to verifying the potential provided by Theorem 4.2. One should note that those LMIs can be solved numerically, providing nice guides for choosing appropriate analytical weights. Symbolic computations and computer algebra software might also help.

The following LMI relies on 6 *dual variables* $\lambda_1, \ldots, \lambda_6$ as it involves gradients and/or function values of $f(\cdot)$ at three points: x_k, x_{k+1}, and $x_\star$, thereby fixing $n = 3$ and hence $n(n-1) = 6$ dual variables.

Theorem C.2. Let $A_{k+1}, A_k \geq 0$ and $\gamma_k \in \mathbb{R}$. The inequality

$$A_{k+1}(f(x_{k+1}) - f_\star) + \tfrac{L}{2}\|x_{k+1} - x_\star\|_2^2 \leq A_k(f(x_k) - f_\star) + \tfrac{L}{2}\|x_k - x_\star\|_2^2$$

holds for all $d \in \mathbb{N}$, all $f \in \mathcal{F}_L(\mathbb{R}^d)$, all $x_k, x_{k+1}, x_\star \in \mathbb{R}^d$ (such that $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$ and $\nabla f(x_\star) = 0$) if and only if

$$\exists \lambda_1, \lambda_2, \ldots, \lambda_6 \geq 0 : \begin{cases} 0 = A_k + \lambda_1 + \lambda_2 - \lambda_4 - \lambda_6 \\ 0 = -A_{k+1} - \lambda_2 + \lambda_3 + \lambda_4 - \lambda_5 \\ 0 \preceq \begin{bmatrix} 0 & \star & \star \\ \frac{1}{2}(\gamma_k L - \lambda_1) & \frac{\lambda_1+\lambda_2+\lambda_4+\lambda_6-\gamma_k^2 L^2 - 2\gamma_k L \lambda_2}{2L} & \star \\ -\frac{\lambda_3}{2} & \frac{1}{2}\left(\gamma_k(\lambda_3+\lambda_4) - \frac{\lambda_2+\lambda_4}{L}\right) & \frac{\lambda_2+\lambda_3+\lambda_4+\lambda_5}{2L} \end{bmatrix}, \end{cases}$$

(where $\star$'s denote symmetric elements in the matrix).

Remark C.1. The LMIs of this section are put in their "raw" forms, for simplicity of the presentation (which does not focus on solving those LMIs analytically. Of course, a few simplifications are relatively direct: for instance, any feasible point will have $\lambda_1 = \gamma_k L$ and $\lambda_3 = 0$, as the corresponding matrix could not be positive semidefinite otherwise.

As we discuss in the sequel (see Remark C.4), it is also relatively straightforward to obtain weaker versions of those LMIs which are then only sufficient for obtaining valid worst-case guarantees. Those simplified LMIs might be simpler to solve analytically, and might therefore be advantageous in certain contexts. Brief discussions and pointers for this topic are provided in Remark C.4 and Section C.5.

A strongly convex version of Theorem C.2 is provided in Theorem C.5. It is slightly more algrebaic in its vanilla form, but allows recovering the results of Theorem 4.10 as a feasible point. Analyses of accelerated methods can be obtained in a similar way, as illustrated by the following LMI. The latter uses on 12 *dual variables* $\lambda_1, \dots, \lambda_{12}$, as it relies on evaluating gradients and/or function values of $f(\cdot)$ at four points: y_k, x_k, x_{k+1}, and $x_\star$, so $n = 4$ and hence $n(n-1) = 12$. Although this LMI might appear as a bit of a brutal approach to worst-case analysis, one might observe that many of elements of the LMI can be set to zero due to the structure of the problem.

Theorem C.3. Let $A_{k+1}, A_k \geq 0$ and $\alpha_k, \gamma_k, \tau_k \in \mathbb{R}$, and consider the iteration

$$\begin{aligned} y_k &= x_k + \tau_k(z_k - x_k) \\ x_{k+1} &= y_k - \alpha_k \nabla f(y_k) \\ z_{k+1} &= z_k - \gamma_k \nabla f(y_k). \end{aligned} \tag{C.3}$$

The inequality

$$A_{k+1}(f(x_{k+1}) - f_\star) + \tfrac{L}{2}\|z_{k+1} - x_\star\|_2^2 \leq A_k(f(x_k) - f_\star) + \tfrac{L}{2}\|z_k - x_\star\|_2^2$$

holds for all $d \in \mathbb{N}$, all $f \in \mathcal{F}_L(\mathbb{R}^d)$, and all $x_k, x_{k+1}, z_k, z_{k+1}, x_\star \in \mathbb{R}^d$ (such that x_{k+1}, z_{k+1} are generated by (C.3) and $\nabla f(x_\star) = 0$) if and only if

$$\exists \lambda_1, \lambda_2, \dots, \lambda_{12} \geq 0 : \begin{cases} 0 = A_k + \lambda_1 + \lambda_2 - \lambda_4 - \lambda_6 - \lambda_8 + \lambda_{11} \\ 0 = -A_{k+1} - \lambda_2 + \lambda_3 + \lambda_4 - \lambda_5 - \lambda_9 + \lambda_{12} \\ 0 = \lambda_7 + \lambda_8 + \lambda_9 - \lambda_{10} - \lambda_{11} - \lambda_{12} \\ 0 \preceq \begin{bmatrix} 0 & 0 & S_{1,3} & S_{1,4} & S_{1,5} \\ 0 & 0 & S_{2,3} & S_{2,4} & S_{2,5} \\ S_{1,3} & S_{2,3} & S_{3,3} & S_{3,4} & S_{3,5} \\ S_{1,4} & S_{2,4} & S_{3,4} & S_{4,4} & S_{4,5} \\ S_{1,5} & S_{2,5} & S_{3,5} & S_{4,5} & S_{5,5} \end{bmatrix}, \end{cases}$$

with

$$
\begin{aligned}
&S_{1,3} = \tfrac{1}{2}(\lambda_7(\tau_k - 1) + \lambda_8\tau_k),\\
&S_{1,4} = -\tfrac{1}{2}(\lambda_1 + \tau_k(\lambda_2 + \lambda_{11})),\ S_{1,5} = \tfrac{1}{2}(\lambda_3(\tau_k - 1) + \lambda_4\tau_k),\\
&S_{2,3} = \tfrac{1}{2}(\gamma_k L - \tau_k(\lambda_7 + \lambda_8)),\\
&S_{2,4} = \tfrac{1}{2}\tau_k(\lambda_2 + \lambda_{11}),\ S_{2,5} = -\tfrac{1}{2}\tau_k(\lambda_3 + \lambda_4),\\
&S_{3,3} = \frac{\lambda_7 + \lambda_8 + \lambda_9 + \lambda_{10} + \lambda_{11} + \lambda_{12} - \gamma_k^2 L^2 - 2\alpha_k L\lambda_9}{2L},\\
&S_{3,4} = -\frac{\alpha_k L\lambda_2 + \lambda_8 + \lambda_{11}}{2L},\ S_{3,5} = \tfrac{1}{2}\left(\alpha_k(\lambda_3 + \lambda_4 + \lambda_{12}) - \frac{\lambda_9 + \lambda_{12}}{L}\right),\\
&S_{4,4} = \frac{\lambda_1 + \lambda_2 + \lambda_4 + \lambda_6 + \lambda_8 + \lambda_{11}}{2L},\ S_{4,5} = -\frac{\lambda_2 + \lambda_4}{2L},\\
&S_{5,5} = \frac{\lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_9 + \lambda_{12}}{2L}.
\end{aligned}
$$

C.3 Analysis of Gradient Descent via Linear Matrix Inequalities

In this section, we detail the approach to obtain LMIs such as those of Theorem C.1, Theorem C.2 and Theorem C.3. We provide full details for gradient descent. The same technique is presented in a more expeditious way for its accelerated versions afterwards.

C.3.1 Linear Convergence of Gradient Descent

We consider gradient descent for minimizing smooth strongly convex functions. For exposition purposes, we investigate a type of one-iteration worst-case convergence guarantee in terms of the distance to the optimum (see Theorem C.1) for gradient descent, of the form:

$$\|x_{k+1} - x_\star\|_2^2 \leq \tau\|x_k - x_\star\|_2^2 \tag{C.4}$$

which are valid for all $d \in \mathbb{N}$, $x_k, x_{k+1}, x_\star \in \mathbb{R}^d$ and all $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ (L-smooth μ-strongly convex function) when $x_{k+1} = x_k - \gamma_k\nabla f(x_k)$ (gradient descent) and $\nabla f(x_\star) = 0$ ($x_\star$ is optimal for f). In this context, we denote by $\tau_\star$ (we omit the dependence on γ_k, μ, and L for convenience) the smallest value τ for which (C.4) is valid. By definition, this value can be formulated as the solution to an optimization problem looking for worst-case scenarios:

$$
\begin{aligned}
\tau_\star \triangleq \max_{\substack{d,f\\ x_k,x_{k+1},x_\star}}\ & \frac{\|x_{k+1} - x_\star\|_2^2}{\|x_k - x_\star\|_2^2}\\
\text{s.t. } & d \in \mathbb{N}, f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)\\
& x_k, x_{k+1}, x_\star \in \mathbb{R}^d\\
& x_{k+1} = x_k - \gamma_k\nabla f(x_k)\\
& \nabla f(x_\star) = 0.
\end{aligned}
\tag{C.5}
$$

As it is, this problem does not look quite practical. However, it actually admits an equivalent formulation as a linear semidefinite program. As a first step for reaching this

formulation, the previous problem can be formulated in an equivalent *sampled* manner. That is, we sample f at the points where the first-order information is explicitly used:

$$\begin{aligned}\tau_\star = \max_{\substack{d\\ f_k,f_\star\\ g_k,,g_\star\\ x_k,x_{k+1},x_\star}} \ & \frac{\|x_{k+1}-x_\star\|_2^2}{\|x_k-x_\star\|_2^2}\\ \text{s.t. } & d\in\mathbb{N}, f_k, f_\star\in\mathbb{R}\\ & x_k, x_{k+1}, x_\star, g_k, g_\star\in\mathbb{R}^d\\ & \exists f\in\mathcal{F}_{\mu,L}: \begin{cases} f_k = f(x_k) \text{ and } g_k=\nabla f(x_k)\\ f_\star = f(x_\star) \text{ and } g_\star=\nabla f(x_\star)\end{cases}\\ & g_\star = 0\\ & x_{k+1} = x_k-\gamma_k g_k,\end{aligned} \tag{C.6}$$

and f is now represented in terms of its samples at $x_\star$ and x_k.

A second stage in this reformulation consists of replacing the existence of a certain $f\in\mathcal{F}_{\mu,L}$ interpolating (or extending) the samples by an equivalent explicit condition provided by the following theorem.

Theorem C.4 ($\mathcal{F}_{\mu,L}$-interpolation, Theorem 4 in (Taylor *et al.*, 2017c))**.** Let $L>\mu\geq 0$, I be an index set and $S=\{(x_i,g_i,f_i)\}_{i\in I}\subseteq\mathbb{R}^d\times\mathbb{R}^d\times\mathbb{R}$ be a set of triplets. There exists $f\in\mathcal{F}_{\mu,L}$ satisfying $f(x_i)=f_i$ and $g_i\in\partial f(x_i)$ for all $i\in I$ if and only if

$$\begin{aligned}f_i \geq f_j + \langle g_j; x_i-x_j\rangle + \frac{1}{2L}\|g_i-g_j\|_2^2\\ + \frac{\mu}{2(1-\mu/L)}\|x_i-x_j-\tfrac{1}{L}(g_i-g_j)\|_2^2\end{aligned} \tag{C.7}$$

holds for all $i,j\in I$.

Theorem C.4 conveniently allows replacing the existence constraint by a set of quadratic inequalities, reaching:

$$\begin{aligned}\tau_\star = \max_{\substack{d\\ f_k,f_\star\\ g_k,x_k,x_\star}} \ & \frac{\|x_k-\gamma_k g_k-x_\star\|_2^2}{\|x_k-x_\star\|_2^2}\\ \text{s.t. } & d\in\mathbb{N}, f_k, f_\star\in\mathbb{R}\\ & x_k, x_\star, g_k\in\mathbb{R}^d\\ & f_\star \geq f_k + \langle g_k; x_\star-x_k\rangle + \frac{1}{2L}\|g_k\|_2^2\\ & \qquad + \frac{\mu}{2(1-\mu/L)}\|x_k-\frac{1}{L}g_k-x_\star\|_2^2\\ & f_k \geq f_\star + \frac{1}{2L}\|g_k\|_2^2\\ & \qquad + \frac{\mu}{2(1-\mu/L)}\|x_k-\frac{1}{L}g_k-x_\star\|_2^2,\end{aligned} \tag{C.8}$$

where we also substituted x_{k+1} and $g_\star$ by their respective expressions. Finally, we arrive to a first (convex) semidefinite reformulation of the problem via new variables: $G \succeq 0$ and F defined as

$$G \triangleq \begin{bmatrix} \|x_k - x_\star\|_2^2 & \langle g_k, x_k - x_\star\rangle \\ \langle g_k, x_k - x_\star\rangle & \|g_k\|_2^2 \end{bmatrix}, \quad F \triangleq f_k - f_\star.$$

The problem turns out to be linear in G and F:

$$\begin{aligned} \tau_\star = \max_{G,F}\ & \frac{G_{1,1} + \gamma_k^2 G_{2,2} - 2\gamma_k G_{1,2}}{G_{1,1}} \\ \text{s.t. } & F \in \mathbb{R}, G \in \mathbb{S}^2 \\ & G \succeq 0 \\ & F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{L}{L-\mu} G_{1,2} \leq 0 \\ & -F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{\mu}{L-\mu} G_{1,2} \leq 0. \end{aligned} \tag{C.9}$$

Finally, a simple homogeneity argument (for any feasible (G, F) to (C.9), the pair $(\tilde{G}, \tilde{F}) \triangleq (G/G_{1,1}, F/G_{1,1})$ is also feasible with the same objective value, with $\tilde{G}_{1,1} = 1$ so we can assume without loss of generality that $G_{1,1} = 1$ without changing the optimal value of the problem—note that it is relatively straightforward to establish that the optimal solution satisfies $G_{1,1} \neq 0$) allows arriving to the equivalent:

$$\begin{aligned} \tau_\star = \max_{G,F}\ & G_{1,1} + \gamma_k^2 G_{2,2} - 2\gamma_k G_{1,2} \\ \text{s.t. } & F \in \mathbb{R}, G \in \mathbb{S}^2 \\ & G \succeq 0 \\ & F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{L}{L-\mu} G_{1,2} \leq 0 \\ & -F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{\mu}{L-\mu} G_{1,2} \leq 0 \\ & G_{1,1} = 1. \end{aligned} \tag{C.10}$$

For arriving to the desired LMI, it remains to dualize the problem. That is, we perform the following primal-dual associations:

$$\begin{aligned} & F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{L}{L-\mu} G_{1,2} \leq 0 && : \lambda_1, \\ & -F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{\mu}{L-\mu} G_{1,2} \leq 0 && : \lambda_2, \\ & G_{1,1} = 1 && : \tau. \end{aligned}$$

Standard Lagrangian duality allows arriving to

$$\begin{aligned} \tau_\star = \min_{\lambda_1,\lambda_2,\tau \geq 0}\ & \tau \\ \text{s.t. } & \lambda_1 = \lambda_2 \\ & 0 \preceq \begin{bmatrix} \tau - 1 + \frac{\mu L(\lambda_1+\lambda_2)}{2(L-\mu)} & \gamma_k - \frac{\lambda_1 L + \lambda_2 \mu}{2(L-\mu)} \\ \gamma_k - \frac{\lambda_1 L + \lambda_2 \mu}{2(L-\mu)} & -\gamma_k^2 + \frac{\lambda_1+\lambda_2}{2(L-\mu)} \end{bmatrix}, \end{aligned} \tag{C.11}$$

where we used the fact there is no duality gap, as one can show the existence of a Slater point (Boyd and Vandenberghe, 2004). One such Slater point can be obtained by applying gradient descent on the function $f(x) = \frac{1}{2}x^\top \mathrm{diag}(L,\mu)x$ (i.e., $d = 2$) with $x_k = (1,1)$. A formal statement is provided in (Taylor *et al.*, 2017c, Theorem 6).

Theorem C.1 is now a direct consequence of the dual reformulation (C.11), as provided by the following proof.

Proof of Theorem C.1. (Sufficiency, $\Leftarrow$) If there exists a feasible point $(\tau, \lambda_1, \lambda_2)$ for (C.2), weak duality implies that it is an upper bound on $\tau_\star$ by construction.

(Necessity, $\Rightarrow$) For any τ such that there exists no $\lambda_1, \lambda_2 \geq 0$ for which $(\tau, \lambda_1, \lambda_2)$ is feasible for (C.2), it follows that $\tau \leq \tau_\star$, and strong duality implies that there exists a problem instance ($f \in \mathcal{F}_{\mu,L}$, $d \in \mathbb{N}$, and $x_k \in \mathbb{R}^d$) on which $\|x_{k+1} - x_\star\|_2^2 = \tau_\star \|x_k - x_\star\|_2^2 \geq \tau \|x_k - x_\star\|_2^2$. ■

Remark C.2. Following similar lines as those of this section, one can verify other types of inequalities, beyond (C.1), simply by changing the objective in (C.5). This allows obtaining the statement from Theorem C.2 and Theorem C.3.

Remark C.3. Finding analytical solutions to such LMIs (parametrized by the algorithm and problem parameters) might be challenging. For gradient descent, the solution is provided in e.g., (Lessard *et al.*, 2016, Section 4.4) and (Taylor *et al.*, 2018b, Theorem 3.1). For more complicated cases, one can rely on numerical inspiration for finding analytical solutions (or upper bounds on it).

Remark C.4. It is possible to obtain "weaker" LMIs based on other sets of inequalities (which are necessary but not sufficient for interpolation). Those LMIs are then only sufficient for finding worst-case guarantees. Those alternate LMIs might enjoy simpler analytical solutions, but this comes at the cost of loosing a priori tightness guarantees. This is in general not a problem if the worst-case guarantee is satisfying, but the subtle consequence is that those LMIs might then fail to provide a satisfying guarantee even when there exists one.

C.3.2 Potential Function for Gradient Descent

For formulating the LMI for verifying potential functions as those of Theorem 4.2 and Theorem 4.10, one essentially has to follow the same steps as in the previous section. The strongly convex version is a bit heavy and is provided below. In short, verifying that

$$\phi_k \triangleq A_k(f(x_k) - f_\star) + \tfrac{L+\mu A_k}{2}\|x_k - x_\star\|_2^2$$

is a potential function, that is, $\phi_{k+1} \leq \phi_k$ (for all $f \in \mathcal{F}_{\mu,L}$, $d \in \mathbb{N}$, and $x_k \in \mathbb{R}^d$), amount to verify that

$$\begin{aligned} 0 \geq \max\{\phi_{k+1} - \phi_k : d \in \mathbb{N},\ f \in \mathcal{F}_{\mu,L},\ x_k, x_{k+1}, x_\star \in \mathbb{R}^d, \\ x_{k+1} = x_k - \gamma_k \nabla f(x_k),\ \text{and } \nabla f(x_\star) = 0\}, \end{aligned}$$

where the maximum is taken over d, f, x_k, x_{k+1} and $x_\star$. This problem can be reformulated as in Section C.3 using the same technique with more samples. More precisely, this formulation requires sampling the function f at three points (instead of two): $x_\star$, x_k, and x_{k+1}, and hence 6 dual variables are required (because 6 inequalities of the form (C.7) are used for describing the sampled version of the function f). The formal statement is provided by the following theorem, without a proof.

Theorem C.5. Let $A_{k+1}, A_k \geq 0$ and $\gamma_k \in \mathbb{R}$. The inequality

$$\begin{aligned} A_{k+1}(f(x_{k+1}) - f_\star) + \tfrac{L+\mu A_{k+1}}{2}\|x_{k+1} - x_\star\|_2^2 \\ \leq A_k(f(x_k) - f_\star) + \tfrac{L+\mu A_k}{2}\|x_k - x_\star\|_2^2 \end{aligned}$$

holds for all $d \in \mathbb{N}$, all $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, all $x_k, x_{k+1}, x_\star \in \mathbb{R}^d$ (such that $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$ and $\nabla f(x_\star) = 0$) if and only if

$$\exists \lambda_1, \lambda_2, \ldots, \lambda_6 \geq 0 : \begin{cases} 0 = A_k + \lambda_1 + \lambda_2 - \lambda_4 - \lambda_6 \\ 0 = -A_{k+1} - \lambda_2 + \lambda_3 + \lambda_4 - \lambda_5 \\ 0 \preceq \begin{bmatrix} S_{1,1} & S_{1,2} & S_{1,3} \\ S_{1,2} & S_{2,2} & S_{2,3} \\ S_{1,3} & S_{2,3} & S_{3,3} \end{bmatrix}, \end{cases}$$

with

$$\begin{aligned} S_{1,1} &= \tfrac{1}{2}\mu\left(A_k - A_{k+1} + \tfrac{L(\lambda_1+\lambda_3+\lambda_5+\lambda_6)}{L-\mu}\right) \\ S_{1,2} &= -\tfrac{\gamma_k(\mu A_{k+1}(\mu-L)+L(\mu(\lambda_3+\lambda_5+1)-L))+\lambda_6\mu+\lambda_1 L}{2(L-\mu)} \\ S_{1,3} &= -\tfrac{\lambda_5\mu+\lambda_3 L}{2(L-\mu)} \\ S_{2,2} &= \tfrac{\gamma_k^2(\mu A_{k+1}(\mu-L)+L(\mu(\lambda_2+\lambda_3+\lambda_4+\lambda_5+1)-L))-2\gamma_k(\lambda_4\mu+\lambda_2 L)+\lambda_1+\lambda_2+\lambda_4+\lambda_6}{2(L-\mu)} \\ S_{2,3} &= \tfrac{\gamma_k(\mu(\lambda_2+\lambda_5)+L(\lambda_3+\lambda_4))-\lambda_2-\lambda_4}{2(L-\mu)} \\ S_{3,3} &= \tfrac{\lambda_2+\lambda_3+\lambda_4+\lambda_5}{2(L-\mu)}. \end{aligned}$$

Note again that a notebook is provided in Section C.5 for obtaining and verifying this LMI formulation via symbolic computations.

C.4 Accelerated Gradient Descent via Linear Matrix Inequalities

We provide the main ideas for formulating the LMI for verifying potential functions as those of Theorem 4.10 and Theorem 4.12. In short, verifying that

$$\phi_k \triangleq A_k(f(x_k) - f_\star) + \tfrac{L+\mu A_k}{2}\|z_k - x_\star\|_2^2$$

is a potential function, that is, $\phi_{k+1} \leq \phi_k$ (for all $f \in \mathcal{F}_{\mu,L}$, $d \in \mathbb{N}$, and $x_k, z_k, x_\star \in \mathbb{R}^d$, $\nabla f(x_\star) = 0$), amounts to verify that

$$\begin{aligned} 0 \geq \max\{\phi_{k+1} - \phi_k : d \in \mathbb{N},\ f \in \mathcal{F}_{\mu,L},\ z_k, x_k, x_\star \in \mathbb{R}^d,\ \nabla f(x_\star) = 0, \\ \text{and } y_k, x_{k+1}, z_{k+1} \in \mathbb{R}^d \text{ generated by (4.17)}\}, \end{aligned}$$

where the maximum is taken over d, f, the iterates, as well as $x_\star$. This problem can be cast as a SDP using the same ideas as in Section C.3 with more samples, again. More precisely, this formulation requires sampling the function f at four points: $x_\star$, x_k, x_{k+1}, and y_k. The case $\mu = 0$ is covered by Theorem C.3.

C.5 Notes and References

General frameworks. The whole idea of using semidefinite programming for analyzing first-order methods dates back to (Drori and Teboulle, 2014) (more details and examples in (Drori, 2014; Drori and Teboulle, 2016)). The principled approach to worst-case analysis using performance estimation problems with interpolation/extension arguments was proposed in in (Taylor *et al.*, 2017c), and generalized to more problem setups in (Taylor *et al.*, 2017a). The integral quadratic approach to first-order methods was proposed in (Lessard *et al.*, 2016), specifically for studying linearly converging methods (focus on strong convexity and related notions). Those two related methodologies were then further extended and linked in different setup (Hu and Lessard, 2017; Hu *et al.*, 2017; Taylor *et al.*, 2018a; Fazlyab *et al.*, 2018; Taylor and Bach, 2019; Lieder, 2021; Aybat *et al.*, 2020; Hu *et al.*, 2021; Ryu and Yin, 2020; Dragomir *et al.*, 2021). Among those developments, some works performed analyses via "weaker" LMIs, based on other sets of inequalities which are necessary but not sufficient for interpolation; see, e.g., (Park and Ryu, 2021). The advantage of this approach is that it is often simpler to obtain analytical solutions to some of those LMIs, at the cost of loosing tightness guarantees (which might not be a problem when the guarantee is satisfying). This is in general the case for IQC-based works. In those cases, non-tightness is usually coupled with the search for a Lyapunov function. In general, it is possible to simultaneously look for a tight guarantee and a Lyapunov/potential function, see e.g., (Taylor *et al.*, 2018a; Taylor and Bach, 2019). A simplified approach to performance estimation problems was implemented in the performance estimation toolbox (Taylor *et al.*, 2017b, PESTO).

Designing methods using semidefinite programming. The optimized gradient method (OGM) was apparently the first method obtained by optimizing its worst-case using SDPs/LMIs. It was obtained as a solution to a convex optimization problem by Drori and Teboulle (2014), which was later solved analytically by Kim and Fessler (2016). The same method was obtained through an analogy with the conjugate gradient method (Drori and Taylor, 2020), which might serve as a strategy for designing method in various setups. Optimized methods can be developed for other criteria and setups as well. As an example, optimized methods for gradient norms $\|\nabla f(x_N)\|_2^2$ are studied in Kim and Fessler (2020) and Kim and Fessler (2018c), in the smooth convex setting. See also Section 4.6.1 and Section 4.6.2; in particular, the *Triple Momentum Method* (TMM) (Van Scoy *et al.*, 2017) was designed as a time-independent optimized gradient method, through Lyapunov arguments (and IQCs). See also (Lessard and Seiler, 2020; Zhou *et al.*, 2020; Gramlich *et al.*, 2020; Drori and Taylor, 2021) for different ways of recovering the TMM. Optimized

methods were also developed in other setups, such as fixed-point iteration (Lieder, 2021) and monotone inclusions (Kim, 2021) (which turned out to be a particular case of (Lieder, 2021)).

Specific methods. The SDP/LMI approaches were taken further for studying first-order methods in a few different contexts. It was originally used for studying gradient-type methods (see, e.g., (Drori and Teboulle, 2014; Drori, 2014; Lessard *et al.*, 2016; Taylor *et al.*, 2017c)) and accelerated/fast gradient-type methods (see, e.g., (Drori and Teboulle, 2014; Drori, 2014; Lessard *et al.*, 2016; Taylor *et al.*, 2017c; Taylor *et al.*, 2017a; Hu and Lessard, 2017; Van Scoy *et al.*, 2017; Cyrus *et al.*, 2018; Safavi *et al.*, 2018; Aybat *et al.*, 2020)) for convex minimization. It was used later for analyzing, among others, nonsmooth setups (Drori and Teboulle, 2016; Drori and Taylor, 2020), stochastic (Hu *et al.*, 2017; Hu *et al.*, 2018; Taylor and Bach, 2019; Hu *et al.*, 2021), coordinate-descent (Shi and Liu, 2017; Taylor and Bach, 2019), nonconvex setups (Abbaszadehpeivasti *et al.*, 2021b; Abbaszadehpeivasti *et al.*, 2021a), proximal methods (Taylor *et al.*, 2017a; Kim and Fessler, 2018b; Kim and Fessler, 2020; Barré *et al.*, 2020a), splitting methods (Ryu and Vũ, 2020; Ryu *et al.*, 2020; Taylor *et al.*, 2018b), monotone inclusions and variational inequalities (Ryu *et al.*, 2020; Gu and Yang, 2019; Gu and Yang, 2020; Zhang *et al.*, 2021), fixed-point iterations (Lieder, 2021), and distributed/decentralized optimization (Sundararajan *et al.*, 2020; Colla and Hendrickx, 2021).

Obtaining and solving the LMIs. For solving the LMIs, standard numerical semidefinite optimization packages can be used, see, e.g., (Lofberg, 2004; Sturm, 1999; Mosek, 2010; Toh *et al.*, 2012). For obtaining and verifying analytical solutions, symbolic computing might also be a great asset. For the purpose of reproducibility, we provide notebooks for obtaining the LMI formulations of this section symbolically, and for solving them numerically, at https://github.com/AdrienTaylor/AccelerationMonograph.

Acknowledgements

The authors would like to warmly thank Raphaël Berthier, Mathieu Barré, Aymeric Dieuleveut, Fabian Pedregosa and Baptiste Goujaud for comments on early versions of this manuscript; for spotting a few typos; and for discussions and developments related to Section 2, Section 4, and Section 5. We are also greatly indebted to Lenaïc Chizat, Laurent Condat, Jelena Diakonikolas, Alexander Gasnikov, Shuvomoy Das Gupta, Pontus Giselsson, Cristóbal Guzmán, Julien Mairal, and Irène Waldspurger for spotting a few typos and inconsistencies in the first version of the manuscript.

We further want to thank Francis Bach, Sébastien Bubeck, Radu-Alexandru Dragomir, Yoel Drori, Hadrien Hendrikx, Reza Babanezhad, Claude Brezinski, Pavel Dvurechensky, Hervé Le Ferrand, Georges Lan, Adam Ouorou, Michela Redivo-Zaglia, Simon Lacoste-Julien, Vincent Roulet, and Ernest Ryu for fruitful discussions and pointers, which largely simplified the writing and revision process of this manuscript.

AA is also extremely grateful to the French ministry of education and école Etienne Marcel for keeping school mostly open during the 2020-2021 pandemic.

AA is at the Département d'informatique de l'ENS, École normale supérieure, UMR CNRS 8548, PSL Research University, 75005 Paris, France and INRIA. AA would like to acknowledge support from the ML and Optimisation joint research initiative with the funds AXA pour la Recherche and Kamet Ventures, a Google focused award, as well as funding from the French government under the management of the Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute). AT is at INRIA and the Département d'informatique de l'ENS, École normale supérieure, CNRS, PSL Research University, 75005 Paris, France. AT acknowledges support from the European Research Council (ERC grant SEQUOIA 724063).

References

Abbaszadehpeivasti, H., E. de Klerk, and M. Zamani. 2021a. "On the rate of convergence of the Difference-of-Convex Algorithm (DCA)". *reprint arXiv:2109.13566*.

Abbaszadehpeivasti, H., E. de Klerk, and M. Zamani. 2021b. "The exact worst-case convergence rate of the gradient method with fixed step lengths for L-smooth functions". *Optimization Letters*.

Aitken, A. C. 1927. "On Bernoulli's Numerical Solution of Algebraic Equations". *Proceedings of the Royal Society of Edinburgh*. 46: 289–305.

Allen-Zhu, Z. 2017. "Katyusha: The first direct acceleration of stochastic gradient methods". *The Journal of Machine Learning Research (JMLR)*. 18(1): 8194–8244.

Allen-Zhu, Z. and L. Orecchia. 2017. "Linear Coupling: An Ultimate Unification of Gradient and Mirror Descent". In: *Proceedings of the 8th Innovations in Theoretical Computer Science Conference (ITCS)*.

Anderson, D. G. 1965. "Iterative procedures for nonlinear integral equations". *Journal of the ACM (JACM)*. 12(4): 547–560.

Anderson, E. J. and P. Nash. 1987. *Linear programming in infinite-dimensional spaces*. Edward J. Anderson, Peter Nash. Chichester: Wiley.

Armijo, L. 1966. "Minimization of functions having Lipschitz continuous first partial derivatives". *Pacific Journal of mathematics*. 16(1): 1–3.

Attouch, H., Z. Chbani, J. Peypouquet, and P. Redont. 2018. "Fast convergence of inertial dynamics and algorithms with asymptotic vanishing viscosity". *Mathematical Programming*. 168(1): 123–175.

Auslender, A. and M. Teboulle. 2006. "Interior gradient and proximal methods for convex and conic optimization". *SIAM Journal on Optimization*. 16(3): 697–725.

Aybat, N. S., A. Fallah, M. Gurbuzbalaban, and A. Ozdaglar. 2019. "A Universally Optimal Multistage Accelerated Stochastic Gradient Method". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Aybat, N. S., A. Fallah, M. Gurbuzbalaban, and A. Ozdaglar. 2020. "Robust accelerated gradient methods for smooth strongly convex functions". *SIAM Journal on Optimization*. 30(1): 717–751.

Baes, M. 2009. "Estimate sequence methods: extensions and approximations". URL: http://www.optimization-online.org/DB_FILE/2009/08/2372.pdf.

Bansal, N. and A. Gupta. 2019. "Potential-function proofs for gradient methods". *Theory of Computing*. 15(1): 1–32.

Barré, M., A. Taylor, and F. Bach. 2020a. "Principled Analyses and Design of First-Order Methods with Inexact Proximal Operators". *arXiv:2006.06041*.

Barré, M., A. Taylor, and F. Bach. 2021. "A note on approximate accelerated forward-backward methods with absolute and relative errors, and possibly strongly convex objectives". *arXiv:2106.15536*.

Barré, M., A. Taylor, and A. d'Aspremont. 2020b. "Convergence of constrained Anderson acceleration". *arXiv:2010.15482*.

Bauschke, H. H., J. Bolte, and M. Teboulle. 2016. "A descent Lemma beyond Lipschitz gradient continuity: first-order methods revisited and applications". *Mathematics of Operations Research*. 42(2): 330–348.

Beck, A. and M. Teboulle. 2009a. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". *SIAM Journal on Imaging Sciences*. 2(1): 183–202.

Beck, A. and M. Teboulle. 2003. "Mirror descent and nonlinear projected subgradient methods for convex optimization". *Operations Research Letters*. 31(3): 167–175.

Beck, A. and M. Teboulle. 2009b. "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems". *IEEE Transactions on Image Processing*. 18(11): 2419–2434.

Becker, S. R., E. J. Candès, and M. C. Grant. 2011. "Templates for convex cone problems with applications to sparse signal recovery". *Mathematical Programming Computation*. 3(3): 165–218.

Ben-Tal, A. and A. S. Nemirovsky. 2001. *Lectures on modern convex optimization : analysis, algorithms, and engineering applications. MPS-SIAM series on optimization*. SIAM.

Bolte, J., A. Daniilidis, and A. Lewis. 2007. "The Lojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems". *SIAM Journal on Optimization*. 17(4): 1205–1223.

Bolte, J., T. P. Nguyen, J. Peypouquet, and B. W. Suter. 2017. "From error bounds to the complexity of first-order descent methods for convex functions". *Mathematical Programming*. 165(2): 471–507.

Bonnans, J.-F., J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal. 2006. *Numerical optimization: theoretical and practical aspects*. Springer Science & Business Media.

Bottou, L. and O. Bousquet. 2007. "The tradeoffs of large scale learning". In: *Advances in Neural Information Processing Systems (NIPS)*.

Boyd, S., N. Parikh, E. Chu, B. Peleato, and J. Eckstein. 2011. "Distributed optimization and statistical learning via the alternating direction method of multipliers". *Foundations and Trends in Machine learning*. 3(1): 1–122.

Boyd, S. and L. Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

Brezinski, C. 1970. "Application de l'ε-algorithme à la résolution des systèmes non linéaires". *Comptes Rendus de l'Académie des Sciences de Paris*. 271(A): 1174–1177.

Brezinski, C. 1971. "Sur un algorithme de résolution des systèmes non linéaires". *Comptes Rendus de l'Académie des Sciences de Paris*. 272(A): 145–148.

Brezinski, C. 1975. "Généralisations de la transformation de Shanks, de la table de Padé et de l'ε-algorithme". *Calcolo*. 12(4): 317–360.

Brezinski, C. 2001. "Convergence acceleration during the 20th century". *Numerical Analysis: Historical Developments in the 20th Century*: 113.

Brezinski, C., S. Cipolla, M. Redivo-Zaglia, and Y. Saad. 2020. "Shanks and Anderson-type acceleration techniques for systems of nonlinear equations". *arXiv:2007.05716*.

Brezinski, C. and M. Redivo–Zaglia. 2019. "The genesis and early developments of Aitken's process, Shanks' transformation, the ε–algorithm, and related fixed point methods". *Numerical Algorithms*. 80(1): 11–133.

Brezinski, C., M. Redivo-Zaglia, and Y. Saad. 2018. "Shanks sequence transformations and Anderson acceleration". *SIAM Review*. 60(3): 646–669.

Brezinski, C. and M. R. Zaglia. 1991. *Extrapolation methods: theory and practice*. Elsevier.

Bubeck, S. 2015. "Convex Optimization: Algorithms and Complexity". *Foundations and Trends in Machine Learning*. 8(3-4): 231–357.

Bubeck, S., Y. T. Lee, and M. Singh. 2015. "A geometric alternative to Nesterov's accelerated gradient descent". *arXiv:1506.08187*.

Cabay, S. and L. Jackson. 1976. "A polynomial extrapolation method for finding limits and antilimits of vector sequences". *SIAM Journal on Numerical Analysis*. 13(5): 734–752.

Calatroni, L. and A. Chambolle. 2019. "Backtracking strategies for accelerated descent methods with smooth composite objectives". *SIAM Journal on Optimization*. 29(3): 1772–1798.

Cauchy, A. 1847. "Méthode générale pour la résolution des systemes d'équations simultanées". *Comptes Rendus de l'Académie des Sciences de Paris*. 25(1847): 536–538.

Chambolle, A. and T. Pock. 2016. "An introduction to continuous optimization for imaging". *Acta Numerica*. 25: 161–319.

Chen, S., S. Ma, and W. Liu. 2017. "Geometric descent method for convex composite minimization". In: *Advances in Neural Information Processing Systems (NIPS)*.

Chierchia, G., E. Chouzenoux, P. L. Combettes, and J.-C. Pesquet. 2020. "The Proximity Operator Repository. User's guide". URL: http://proximity-operator.net/.

Clarke, F. H. 1990. *Optimization and nonsmooth analysis*. Vol. 5. SIAM.

Cohen, A., W. Dahmen, and R. DeVore. 2009. "Compressed sensing and best k-term approximation". *Journal of the AMS*. 22(1): 211–231.

Colla, S. and J. M. Hendrickx. 2021. "Automated Worst-Case Performance Analysis of Decentralized Gradient Descent". In: *Proceedings of the 60th Conference on Decision and Control (CDC)*.

Condat, L., D. Kitahara, A. Contreras, and A. Hirabayashi. 2019. "Proximal splitting algorithms: A tour of recent advances, with new twists". *arXiv:1912.00137*.

Cyrus, S., B. Hu, B. Van Scoy, and L. Lessard. 2018. "A robust accelerated optimization algorithm for strongly convex functions". In: *Proceedings of the 2018 American Control Conference (ACC)*.

d'Aspremont, A. 2008. "Smooth Optimization with Approximate Gradient". *SIAM Journal on Optimization*. 19(3): 1171–1183.

d'Aspremont, A., C. Guzman, and M. Jaggi. 2018. "Optimal affine-invariant smooth minimization algorithms". *SIAM Journal on Optimization*. 28(3): 2384–2405.

Davis, D., D. Drusvyatskiy, and V. Charisopoulos. 2019. "Stochastic algorithms with geometric step decay converge linearly on sharp functions". *arXiv:1907.09547*.

De Klerk, E., F. Glineur, and A. B. Taylor. 2020. "Worst-case convergence analysis of inexact gradient and Newton methods through semidefinite programming performance estimation". *SIAM Journal on Optimization*. 30(3): 2053–2082.

De Klerk, E., F. Glineur, and A. B. Taylor. 2017. "On the worst-case complexity of the gradient method with exact line search for smooth strongly convex functions". *Optimization Letters*. 11(7): 1185–1199.

Defazio, A., F. Bach, and S. Lacoste-Julien. 2014a. "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives". In: *Advances in Neural Information Processing Systems (NIPS)*.

Defazio, A. J., T. S. Caetano, and J. Domke. 2014b. "Finito: A faster, permutable incremental gradient method for big data problems". In: *Proceedings of the 31st International Conference on Machine Learning (ICML)*.

Devolder, O. 2011. "Stochastic first order methods in smooth convex optimization". *Tech. rep*. CORE discussion paper.

Devolder, O. 2013. "Exactness, inexactness and stochasticity in first-order methods for large-scale convex optimization". *PhD thesis*.

Devolder, O., F. Glineur, and Y. Nesterov. 2013. "Intermediate gradient methods for smooth convex problems with inexact oracle". *Tech. rep*. CORE discussion paper.

Devolder, O., F. Glineur, and Y. Nesterov. 2014. "First-order methods of smooth convex optimization with inexact oracle". *Mathematical Programming*. 146(1-2): 37–75.

Diakonikolas, J. and C. Guzmán. 2021. "Complementary Composite Minimization, Small Gradients in General Norms, and Applications to Regression Problems". *arXiv:2101.11041*.

Diakonikolas, J. and L. Orecchia. 2019a. "Conjugate gradients and accelerated methods unified: The approximate duality gap view". *arXiv:1907.00289*.

Diakonikolas, J. and L. Orecchia. 2019b. "The approximate duality gap technique: A unified theory of first-order methods". *SIAM Journal on Optimization*. 29(1): 660–689.

Diakonikolas, J. and P. Wang. 2021. "Potential Function-based Framework for Making the Gradients Small in Convex and Min-Max Optimization". *arXiv:2101.12101*.

Douglas, J. and H. H. Rachford. 1956. "On the numerical solution of heat conduction problems in two and three space variables". *Transactions of the American mathematical Society*. 82(2): 421–439.

Dragomir, R.-A., A. B. Taylor, A. d'Aspremont, and J. Bolte. 2021. "Optimal complexity and certification of Bregman first-order methods". *Mathematical Programming*: 1–43.

Drori, Y. 2014. "Contributions to the Complexity Analysis of Optimization Algorithms". *PhD thesis*. Tel-Aviv University.

Drori, Y. 2017. "The exact information-based complexity of smooth convex minimization". *Journal of Complexity*. 39: 1–16.

Drori, Y. 2018. "On the properties of convex functions over open sets". *arXiv:1812.02419*.

Drori, Y. and A. Taylor. 2021. "On the oracle complexity of smooth strongly convex minimization". *Journal of Complexity*.

Drori, Y. and A. B. Taylor. 2020. "Efficient first-order methods for convex minimization: a constructive approach". *Mathematical Programming*. 184(1): 183–220.

Drori, Y. and M. Teboulle. 2014. "Performance of first-order methods for smooth convex minimization: a novel approach". *Mathematical Programming*. 145(1-2): 451–482.

Drori, Y. and M. Teboulle. 2016. "An optimal variant of Kelley's cutting-plane method". *Mathematical Programming*. 160(1-2): 321–351.

Drusvyatskiy, D., M. Fazel, and S. Roy. 2018. "An optimal first order method based on optimal quadratic averaging". *SIAM Journal on Optimization*. 28(1): 251–271.

Dvurechensky, P. and A. Gasnikov. 2016. "Stochastic intermediate gradient method for convex problems with stochastic inexact oracle". *Journal of Optimization Theory and Applications*. 171(1): 121–145.

Eckstein, J. 1989. "Splitting methods for monotone operators with applications to parallel optimization". *PhD thesis*. Massachusetts Institute of Technology.

Eckstein, J. and P. J. Silva. 2013. "A practical relative error criterion for augmented Lagrangians". *Mathematical Programming*. 141(1-2): 319–348.

Eckstein, J. and W. Yao. 2012. "Augmented Lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results". *RUTCOR Research Reports*. 32(3).

Eddy, R. 1979. "Extrapolating to the limit of a vector sequence". In: *Information linkage between applied mathematics and industry*. Elsevier. 387–396.

Even, M., R. Berthier, F. Bach, N. Flammarion, P. Gaillard, H. Hendrikx, L. Massoulié, and A. Taylor. 2021. "A Continuized View on Nesterov Acceleration for Stochastic Gradient Descent and Randomized Gossip". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Fang, H.-R. and Y. Saad. 2009. "Two classes of multisecant methods for nonlinear acceleration". *Numerical Linear Algebra with Applications*. 16(3): 197–221.

Fazlyab, M., A. Ribeiro, M. Morari, and V. M. Preciado. 2018. "Analysis of optimization algorithms via integral quadratic constraints: Nonstrongly convex problems". *SIAM Journal on Optimization*. 28(3): 2654–2689.

Fercoq, O. and P. Richtárik. 2015. "Accelerated, parallel, and proximal coordinate descent". *SIAM Journal on Optimization*. 25(4): 1997–2023.

Fessler, J. A. 2020. "Optimization methods for magnetic resonance image reconstruction: Key models and optimization algorithms". *IEEE Signal Processing Magazine*. 37(1): 33–40. Complete version: http://arxiv.org/abs/1903.03510.

Fischer, B. 1996. "Polynomial Based Iteration Methods for Symmetric Linear Systems".

Flanders, D. A. and G. Shortley. 1950. "Numerical determination of fundamental modes". *Journal of Applied Physics*. 21(12): 1326–1332.

Florea, M. I. and S. A. Vorobyov. 2018. "An accelerated composite gradient method for large-scale composite objective problems". *IEEE Transactions on Signal Processing*. 67(2): 444–459.

Florea, M. I. and S. A. Vorobyov. 2020. "A generalized accelerated composite gradient method: Uniting Nesterov's fast gradient method and FISTA". *IEEE Transactions on Signal Processing*.

Ford, W. F. and A. Sidi. 1988. "Recursive algorithms for vector extrapolation methods". *Applied numerical mathematics*. 4(6): 477–489.

Gasnikov, A., P. Dvurechensky, E. Gorbunov, E. Vorontsova, D. Selikhanovych, and C. Uribe. 2019. "Optimal Tensor Methods in Smooth Convex and Uniformly Convex Optimization". In: *Proceedings of the 32nd Conference on Learning Theory (COLT)*.

Gasnikov, A. V. and Y. Nesterov. 2018. "Universal method for stochastic composite optimization problems". *Computational Mathematics and Mathematical Physics*. 58(1): 48–64.

Gekeler, E. 1972. "On the solution of systems of equations by the epsilon algorithm of Wynn". *Mathematics of Computation*. 26(118): 427–436.

Glowinski, R. and A. Marroco. 1975. "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires". *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*. 9(R2): 41–76.

Goldstein, A. 1962. "Cauchy's method of minimization". *Numerische Mathematik*. 4(1): 146–150.

Golub, G. H. and R. S. Varga. 1961a. "Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods". *Numerische Mathematik*. 3(1): 147–156.

Golub, G. H. and R. S. Varga. 1961b. "Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods". *Numerische Mathematik*. 3(1): 157–168.

Gorbunov, E., M. Danilova, and A. Gasnikov. 2020. "Stochastic Optimization with Heavy-Tailed Noise via Accelerated Gradient Clipping". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Gramlich, D., C. Ebenbauer, and C. W. Scherer. 2020. "Convex Synthesis of Accelerated Gradient Algorithms for Optimization and Saddle Point Problems using Lyapunov functions". *arXiv:2006.09946*.

Gu, G. and J. Yang. 2019. "On the optimal ergodic sublinear convergence rate of the relaxed proximal point algorithm for variational inequalities". *arXiv:1905.06030*.

Gu, G. and J. Yang. 2020. "Tight sublinear convergence rate of the proximal point algorithm for maximal monotone inclusion problems". *SIAM Journal on Optimization*. 30(3): 1905–1921.

Güler, O. 1991. "On the convergence of the proximal point algorithm for convex minimization". *SIAM Journal on Control and Optimization*. 29(2): 403–419.

Güler, O. 1992. "New proximal point algorithms for convex minimization". *SIAM Journal on Optimization*. 2(4): 649–664.

Gutknecht, M. H. and S. Röllin. 2002. "The Chebyshev iteration revisited". *Parallel Computing*. 28(2): 263–283.

Gutman, D. H. and J. F. Peña. 2018. "A unified framework for Bregman proximal methods: subgradient, gradient, and accelerated gradient schemes". *arXiv:1812.10198*.

Guzmán, C. and A. S. Nemirovsky. 2015. "On lower complexity bounds for large-scale smooth convex optimization". *Journal of Complexity*. 31(1): 1–14.

Hanzely, F., P. Richtarik, and L. Xiao. 2021. "Accelerated Bregman proximal gradient methods for relatively smooth convex optimization". *Computational Optimization and Applications*. 79(2): 405–440.

Hinder, O., A. Sidford, and N. Sohoni. 2020. "Near-Optimal Methods for Minimizing Star-Convex Functions and Beyond". In: *Proceedings of the 33rd Conference on Learning Theory (COLT)*.

Hiriart-Urruty, J.-B. and C. Lemaréchal. 2013. *Convex analysis and minimization algorithms I: Fundamentals*. Vol. 305. Springer science & business media.

Hu, B. and L. Lessard. 2017. "Dissipativity theory for Nesterov's accelerated method". In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*.

Hu, B., P. Seiler, and L. Lessard. 2021. "Analysis of biased stochastic gradient descent using sequential semidefinite programs". *Mathematical Programming*. 187(1): 383–408.

Hu, B., P. Seiler, and A. Rantzer. 2017. "A unified analysis of stochastic optimization methods using jump system theory and quadratic constraints". In: *Proceedings of the 30th Conference on Learning Theory (COLT)*.

Hu, B., S. Wright, and L. Lessard. 2018. "Dissipativity theory for accelerating stochastic variance reduction: A unified analysis of SVRG and Katyusha using semidefinite programs". In: *Proceedings of the 35th International Conference on Machine Learning (ICML)*.

Hu, C., W. Pan, and J. Kwok. 2009. "Accelerated gradient methods for stochastic optimization and online learning". In: *Advances in Neural Information Processing Systems (NIPS)*.

Ito, M. and M. Fukuda. 2021. "Nearly optimal first-order methods for convex optimization under gradient norm measure: An adaptive regularization approach". *Journal of Optimization Theory and Applications*: 1–35.

Iusem, A. N. 1999. "Augmented Lagrangian methods and proximal point methods for convex optimization". *Investigación Operativa*. 8(11-49): 7.

Ivanova, A., D. Grishchenko, A. Gasnikov, and E. Shulgin. 2019. "Adaptive Catalyst for smooth convex optimization". *arXiv:1911.11271*.

Jbilou, K. and H. Sadok. 2000. "Vector extrapolation methods. Applications and numerical comparison". *Journal of Computational and Applied Mathematics*. 122(1-2): 149–165.

Jbilou, K. and H. Sadok. 1991. "Some results about vector extrapolation methods and related fixed-point iterations". *Journal of Computational and Applied Mathematics*. 36(3): 385–398.

Jbilou, K. and H. Sadok. 1995. "Analysis of some vector extrapolation methods for solving systems of linear equations". *Numerische Mathematik*. 70(1): 73–89.

Johnson, R. and T. Zhang. 2013. "Accelerating stochastic gradient descent using predictive variance reduction". In: *Advances in Neural Information Processing Systems (NIPS)*.

Juditsky, A., G. Lan, A. S. Nemirovsky, and A. Shapiro. 2009. "Stochastic Approximation Approach to Stochastic Programming". *SIAM Journal on Optimization*. 19(4): 1574–1609.

Juditsky, A. and A. S. Nemirovsky. 2011a. "First order methods for nonsmooth convex large-scale optimization, i: general purpose methods". *Optimization for Machine Learning*. 30(9): 121–148.

Juditsky, A. and A. S. Nemirovsky. 2011b. "First order methods for nonsmooth convex large-scale optimization, ii: utilizing problems structure". *Optimization for Machine Learning*. 30(9): 149–183.

Juditsky, A. and Y. Nesterov. 2014. "Deterministic and stochastic primal-dual subgradient algorithms for uniformly convex minimization". *Stochastic Systems*. 4(1): 44–80.

Karimi, S. and S. Vavasis. 2017. "A single potential governing convergence of conjugate gradient, accelerated gradient and geometric descent". *arXiv:1712.09498*.

Karimi, S. and S. A. Vavasis. 2016. "A unified convergence bound for conjugate gradient and accelerated gradient". *arXiv:1605.00320*.

Kerdreux, T., A. d'Aspremont, and S. Pokutta. 2019. "Restarting Frank-Wolfe". In: *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Kim, D. 2021. "Accelerated proximal point method for maximally monotone operators". *Mathematical Programming*: 1–31.

Kim, D. and J. A. Fessler. 2018a. "Adaptive restart of the optimized gradient method for convex optimization". *Journal of Optimization Theory and Applications*. 178(1): 240–263.

Kim, D. and J. A. Fessler. 2016. "Optimized first-order methods for smooth convex minimization". *Mathematical Programming*. 159(1-2): 81–107.

Kim, D. and J. A. Fessler. 2017. "On the convergence analysis of the optimized gradient method". *Journal of Optimization Theory and Applications*. 172(1): 187–205.

Kim, D. and J. A. Fessler. 2018b. "Another look at the fast iterative shrinkage/thresholding algorithm (FISTA)". *SIAM Journal on Optimization*. 28(1): 223–250.

Kim, D. and J. A. Fessler. 2018c. "Generalizing the optimized gradient method for smooth convex minimization". *SIAM Journal on Optimization*. 28(2): 1920–1950.

Kim, D. and J. A. Fessler. 2020. "Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions". *Journal of Optimization Theory and Applications*.

Krichene, W., A. Bayen, and P. L. Bartlett. 2015. "Accelerated mirror descent in continuous and discrete time". In: *Advances in Neural Information Processing Systems (NIPS)*.

Kulunchakov, A. and J. Mairal. 2019. "A Generic Acceleration Framework for Stochastic Composite Optimization". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Kulunchakov, A. and J. Mairal. 2020. "Estimate Sequences for Stochastic Composite Optimization: Variance Reduction, Acceleration, and Robustness to Noise". *The Journal of Machine Learning Research (JMLR)*. 21(155): 1–52.

Kurdyka, K. 1998. "On gradients of functions definable in o-minimal structures". In: *Annales de l'institut Fourier*. Vol. 48. No. 3. 769–783.

Lacoste-Julien, S. and M. Jaggi. 2015. "On the global linear convergence of Frank-Wolfe optimization variants". In: *Advances in Neural Information Processing Systems (NIPS)*.

Lacotte, J. and M. Pilanci. 2020. "Optimal randomized first-order methods for least-squares problems". In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*.

Lan, G. 2008. "Efficient Methods for stochastic composite optimization". *Tech. rep.* School of Industrial and Systems Engineering, Georgia Institute of Technology. URL: http://www.optimization-online.org/DB_HTML/2008/08/2061.html.

Lan, G. 2012. "An optimal method for stochastic composite optimization". *Mathematical Programming*. 133(1-2): 365–397.

Lan, G., Z. Lu, and R. D. Monteiro. 2011. "Primal-dual first-order methods with $O(1/\epsilon)$ iteration-complexity for cone programming". *Mathematical Programming*. 126(1): 1–29.

Lee, J., C. Park, and E. K. Ryu. 2021. "A Geometric Structure of Acceleration and Its Role in Making Gradients Small Fast". *arXiv:2106.10439*.

Lee, Y. T. and A. Sidford. 2013. "Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems". In: *54th Symposium on Foundations of Computer Science*. 147–156.

Lemaréchal, C. and C. Sagastizábal. 1997. "Practical Aspects of the Moreau–Yosida Regularization: Theoretical Preliminaries". *SIAM Journal on Optimization*. 7(2): 367–385.

Lessard, L., B. Recht, and A. Packard. 2016. "Analysis and design of optimization algorithms via integral quadratic constraints". *SIAM Journal on Optimization*. 26(1): 57–95.

Lessard, L. and P. Seiler. 2020. "Direct synthesis of iterative algorithms with bounds on achievable worst-case convergence rate". In: *Proceedings of the 2020 American Control Conference (ACC)*.

Li, G. and T. K. Pong. 2018. "Calculus of the exponent of Kurdyka–Łojasiewicz inequality and its applications to linear convergence of first-order methods". *Foundations of computational mathematics*. 18(5): 1199–1232.

Lieder, F. 2021. "On the convergence rate of the Halpern-iteration". *Optimization Letters*. 15(2): 405–418.

Lin, H., J. Mairal, and Z. Harchaoui. 2015. "A universal catalyst for first-order optimization". In: *Advances in Neural Information Processing Systems (NIPS)*.

Lin, H., J. Mairal, and Z. Harchaoui. 2018. "Catalyst acceleration for first-order convex optimization: from theory to practice". *The Journal of Machine Learning Research (JMLR)*. 18(1): 7854–7907.

Lions, P.-L. and B. Mercier. 1979. "Splitting algorithms for the sum of two nonlinear operators". *SIAM Journal on Numerical Analysis*. 16(6): 964–979.

Lofberg, J. 2004. "YALMIP: A toolbox for modeling and optimization in MATLAB". In: *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*. IEEE. 284–289.

Lojasiewicz, S. 1963. "Une propriété topologique des sous-ensembles analytiques réels". *Les équations aux dérivées partielles*: 87–89.

Lu, H., R. M. Freund, and Y. Nesterov. 2018. "Relatively smooth convex optimization by first-order methods, and applications". *SIAM Journal on Optimization*. 28(1): 333–354.

Luo, Z.-Q. and P. Tseng. 1992. "On the linear convergence of descent methods for convex essentially smooth minimization". *SIAM Journal on Control and Optimization*. 30(2): 408–425.

Mai, V. and M. Johansson. 2020. "Anderson acceleration of proximal gradient methods". In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*.

Mairal, J. 2015. "Incremental majorization-minimization optimization with application to large-scale machine learning". *SIAM Journal on Optimization*. 25(2): 829–855.

Mairal, J. 2019. "Cyanure: An Open-Source Toolbox for Empirical Risk Minimization for Python, C++, and soon more". *arXiv:1912.08165*.

Malitsky, Y. and K. Mishchenko. 2020. "Adaptive Gradient Descent without Descent". In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*.

Martinet, B. 1970. "Régularisation d'inéquations variationnelles par approximations successives". *Revue Française d'Informatique et de Recherche Opérationnelle*. 4: 154–158.

Martinet, B. 1972. "Détermination approchée d'un point fixe d'une application pseudo-contractante. Cas de l'application prox." *Comptes Rendus de l'Académie des Sciences de Paris*. 274: 163–165.

Mason, J. C. and D. C. Handscomb. 2002. *Chebyshev polynomials*. CRC press.

Megretski, A. and A. Rantzer. 1997. "System analysis via integral quadratic constraints". *IEEE Transactions on Automatic Control*. 42(6): 819–830.

Mešina, M. 1977. "Convergence acceleration for the iterative solution of the equations $X = AX + f$". *Computer Methods in Applied Mechanics and Engineering*. 10(2): 165–173.

Mifflin, R. 1977. "Semismooth and semiconvex functions in constrained optimization". *SIAM Journal on Control and Optimization*. 15(6): 959–972.

Monteiro, R. D. and B. F. Svaiter. 2013. "An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods". *SIAM Journal on Optimization*. 23(2): 1092–1125.

Moreau, J.-J. 1962. "Fonctions convexes duales et points proximaux dans un espace hilbertien". *Comptes Rendus de l'Académie des Sciences de Paris*. 255: 2897–2899.

Moreau, J.-J. 1965. "Proximité et dualité dans un espace hilbertien". *Bulletin de la Société mathématique de France*. 93: 273–299.

Mosek, A. 2010. "The MOSEK optimization software". URL: http://www.mosek.com.

Narkiss, G. and M. Zibulevsky. 2005. *Sequential subspace optimization method for large-scale unconstrained problems*. Technion-IIT, Department of Electrical Engineering.

Necoara, I., Y. Nesterov, and F. Glineur. 2019. "Linear convergence of first order methods for non-strongly convex optimization". *Mathematical Programming*. 175(1-2): 69–107.

Nemirovsky, A. S. and D. Yudin. 1983a. *Problem complexity and method efficiency in optimization.*

Nemirovsky, A. S. 1982. "Orth-method for smooth convex optimization". *Izvestia AN SSSR, Transl.: Eng. Cybern. Soviet J. Comput. Syst. Sci.* 2: 937–947.

Nemirovsky, A. S. 1991. "On optimality of Krylov's information when solving linear operator equations". *Journal of Complexity*. 7(2): 121–130.

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

Nemirovsky, A. S. 1994. "Information-based complexity of convex programming". Lecture notes.

Nemirovsky, A. S. and Y. Nesterov. 1985. "Optimal methods of smooth convex minimization". *USSR Computational Mathematics and Mathematical Physics*. 25(2): 21–30.

Nemirovsky, A. S. and B. T. Polyak. 1984. "Iterative methods for solving linear ill-posed problems under precise information." *ENG. CYBER.* (4): 50–56.

Nemirovsky, A. S. and D. B. Yudin. 1983b. "Information-based complexity of mathematical programming (in Russian)". *Izvestia AN SSSR, Ser. Tekhnicheskaya Kibernetika (the journal is translated to English as Engineering Cybernetics. Soviet J. Computer & Systems Sci.)* 1.

Nemirovsky, A. S. and D. B. Yudin. 1983c. "Problem Complexity and Method Efficiency in Optimization." *Willey-Interscience, New York*.

Nesterov, Y. 1983. "A method of solving a convex programming problem with convergence rate $O(1/k^2)$". *Soviet Mathematics Doklady*. 27(2): 372–376.

Nesterov, Y. 2003. *Introductory Lectures on Convex Optimization*. Springer.

Nesterov, Y. 2005. "Smooth minimization of non-smooth functions". *Mathematical Programming*. 103(1): 127–152.

Nesterov, Y. 2009. "Primal-dual subgradient methods for convex problems". *Mathematical programming Series B*. 120(1): 221–259.

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

Nesterov, Y. 2015. "Universal gradient methods for convex optimization problems". *Mathematical Programming*. 152(1-2): 381–404.

Nesterov, Y. 2008. "Accelerating the cubic regularization of Newton's method on convex problems". *Mathematical Programming*. 112(1): 159–181.

Nesterov, Y. 2012a. "Efficiency of coordinate descent methods on huge-scale optimization problems". *SIAM Journal on Optimization*. 22(2): 341–362.

Nesterov, Y. 2012b. "How to make the gradients small". *Optima. Mathematical Optimization Society Newsletter*. (88): 10–11.

Nesterov, Y. 2019. "Implementable tensor methods in unconstrained convex optimization". *Mathematical Programming*: 1–27.

Nesterov, Y. 2020a. "Inexact accelerated high-order proximal-point methods". *Tech. rep.* CORE discussion paper.

Nesterov, Y. 2020b. "Inexact high-order proximal-point methods with auxiliary search procedure". *Tech. rep.* CORE discussion paper.

Nesterov, Y. and B. T. Polyak. 2006. "Cubic regularization of Newton method and its global performance". *Mathematical Programming*. 108(1): 177–205.

Nesterov, Y. and V. Shikhman. 2015. "Quasi-monotone subgradient methods for non-smooth convex minimization". *Journal of Optimization Theory and Applications*. 165(3): 917–940.

Nesterov, Y. and S. U. Stich. 2017. "Efficiency of the accelerated coordinate descent method on structured optimization problems". *SIAM Journal on Optimization*. 27(1): 110–123.

Nocedal, J. and S. Wright. 2006. *Numerical optimization*. Springer Science & Business Media.

O'Donoghue, B. and E. Candes. 2015. "Adaptive restart for accelerated gradient schemes". *Foundations of computational mathematics*. 15(3): 715–732.

Paige, C. C. and M. A. Saunders. 1975. "Solution of sparse indefinite systems of linear equations". *SIAM journal on numerical analysis*. 12(4): 617–629.

Pang, J.-S. 1987. "A posteriori error bounds for the linearly-constrained variational inequality problem". *Mathematics of Operations Research*. 12(3): 474–484.

Paquette, C., H. Lin, D. Drusvyatskiy, J. Mairal, and Z. Harchaoui. 2018. "Catalyst for Gradient-based Nonconvex Optimization". In: *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Parikh, N. and S. Boyd. 2014. "Proximal algorithms". *Foundations and Trends in Optimization*. 1(3): 127–239.

Park, C., J. Park, and E. K. Ryu. 2021. "Factor-$\sqrt{2}$ Acceleration of Accelerated Gradient Methods". *arXiv:2102.07366*.

Park, C. and Ryu. 2021. "Optimal First-Order Algorithms as a Function of Inequalities". *arXiv:2110.11035*.

Passty, G. B. 1979. "Ergodic convergence to a zero of the sum of monotone operators in Hilbert space". *Journal of Mathematical Analysis and Applications*. 72(2): 383–390.

Pedregosa, F. and D. Scieur. 2020. "Acceleration through spectral density estimation". In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*.

Peyré, G. 2011. "The numerical tours of signal processing-advanced computational signal and image processing". *IEEE Computing in Science and Engineering*. 13(4): 94–97.

Qi, L. and J. Sun. 1993. "A nonsmooth version of Newton's method". *Mathematical programming*. 58(1-3): 353–367.

Robbins, H. and S. Monro. 1951. "A stochastic approximation method". *The annals of mathematical statistics*: 400–407.

Rockafellar, R. T. 1973. "A dual approach to solving nonlinear programming problems by unconstrained optimization". *Mathematical Programming*. 5(1): 354–373.

Rockafellar, R. T. 1970. *Convex Analysis*. Princeton.: Princeton University Press.

Rockafellar, R. T. 1976. "Augmented Lagrangians and applications of the proximal point algorithm in convex programming". *Mathematics of operations research*. 1(2): 97–116.

Rockafellar, R. T. and R. J.-B. Wets. 2009. *Variational analysis*. Vol. 317. Springer Science & Business Media.

Roulet, V. and A. d'Aspremont. 2017. "Sharpness, restart and acceleration". In: *Advances in Neural Information Processing Systems (NIPS)*.

Ryu, E. K. and S. Boyd. 2016. "Primer on monotone operator methods". *Appl. Comput. Math.* 15(1): 3–43.

Ryu, E. K., A. B. Taylor, C. Bergeling, and P. Giselsson. 2020. "Operator splitting performance estimation: Tight contraction factors and optimal parameter selection". *SIAM Journal on Optimization*. 30(3): 2251–2271.

Ryu, E. K. and B. C. Vũ. 2020. "Finding the forward-Douglas–Rachford-forward method". *Journal of Optimization Theory and Applications*. 184(3): 858–876.

Ryu, E. K. and W. Yin. 2020. *Large-Scale Convex Optimization via Monotone Operators*.

Saad, Y. and M. H. Schultz. 1986. "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems". *SIAM Journal on scientific and statistical computing*. 7(3): 856–869.

Safavi, S., B. Joshi, G. França, and J. Bento. 2018. "An Explicit Convergence Rate for Nesterov's Method from SDP". In: *IEEE International Symposium on Information Theory (ISIT)*. 1560–1564.

Salzo, S. and S. Villa. 2012. "Inexact and accelerated proximal point algorithms". *Journal of Convex analysis*. 19(4): 1167–1192.

Sanz Serna, J. M. and K. C. Zygalakis. 2021. "The connections between Lyapunov functions for some optimization algorithms and differential equations". *SIAM Journal on Numerical Analysis*. 59(3): 1542–1565.

Scheinberg, K., D. Goldfarb, and X. Bai. 2014. "Fast first-order methods for composite convex optimization with backtracking". *Foundations of Computational Mathematics*. 14(3): 389–417.

Schmidt, M., N. Le Roux, and F. Bach. 2011. "Convergence rates of inexact proximal-gradient methods for convex optimization". In: *Advances in Neural Information Processing Systems (NIPS)*.

Schmidt, M., N. Le Roux, and F. Bach. 2017. "Minimizing finite sums with the stochastic average gradient". *Mathematical Programming*. 162(1-2): 83–112.

Scieur, D., F. Bach, and A. d'Aspremont. 2017a. "Nonlinear acceleration of stochastic algorithms". In: *Advances in Neural Information Processing Systems (NIPS)*.

Scieur, D., A. d'Aspremont, and F. Bach. 2016. "Regularized nonlinear acceleration". In: *Advances in Neural Information Processing Systems (NIPS)*.

Scieur, D., E. Oyallon, A. d'Aspremont, and F. Bach. 2018. "Online Regularized Nonlinear Acceleration". *arXiv:1805.09639*.

Scieur, D. and F. Pedregosa. 2020. "Universal Asymptotic Optimality of Polyak Momentum". In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*.

Scieur, D., V. Roulet, F. Bach, and A. d'Aspremont. 2017b. "Integration methods and optimization algorithms". In: *Advances in Neural Information Processing Systems (NIPS)*.

Shalev-Shwartz, S. and T. Zhang. 2013. "Stochastic dual coordinate ascent methods for regularized loss minimization". *The Journal of Machine Learning Research (JMLR)*. 14: 567–599.

Shalev-Shwartz, S. and T. Zhang. 2014. "Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization". In: *Proceedings of the 31st International Conference on Machine Learning (ICML)*.

Shi, B., S. S. Du, M. I. Jordan, and W. J. Su. 2021. "Understanding the acceleration phenomenon via high-resolution differential equations". *Mathematical Programming*: 1–70.

Shi, B., S. S. Du, W. Su, and M. I. Jordan. 2019. "Acceleration via Symplectic Discretization of High-Resolution Differential Equations". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Shi, Z. and R. Liu. 2017. "Better worst-case complexity analysis of the block coordinate descent method for large scale machine learning". In: *16th IEEE International Conference on Machine Learning and Applications (ICMLA)*.

Sidi, A. 1986. "Convergence and stability properties of minimal polynomial and reduced rank extrapolation algorithms". *SIAM Journal on Numerical Analysis*. 23(1): 197–209.

Sidi, A. 1988. “Extrapolation vs. projection methods for linear systems of equations”. *Journal of Computational and Applied Mathematics*. 22(1): 71–88.

Sidi, A. 1991. “Efficient implementation of minimal polynomial and reduced rank extrapolation methods”. *Journal of Computational and Applied Mathematics*. 36(3): 305–337.

Sidi, A. 2008. “Vector extrapolation methods with applications to solution of large systems of equations and to PageRank computations”. *Computers & Mathematics with Applications*. 56(1): 1–24.

Sidi, A. 2017a. “Minimal polynomial and reduced rank extrapolation methods are related”. *Advances in Computational Mathematics*. 43(1): 151–170.

Sidi, A. 2017b. *Vector extrapolation methods with applications*. SIAM.

Sidi, A. 2019. “A convergence study for reduced rank extrapolation on nonlinear systems”. *Numerical Algorithms*: 1–26.

Sidi, A. and J. Bridger. 1988. “Convergence and stability analyses for some vector extrapolation methods in the presence of defective iteration matrices”. *Journal of Computational and Applied Mathematics*. 22(1): 35–61.

Sidi, A., W. F. Ford, and D. A. Smith. 1986. “Acceleration of convergence of vector sequences”. *SIAM Journal on Numerical Analysis*. 23(1): 178–196.

Sidi, A. and Y. Shapira. 1998. “Upper bounds for convergence rates of acceleration methods with initial iterations”. *Numerical Algorithms*. 18(2): 113–132.

Siegel, J. W. 2019. “Accelerated first-order methods: Differential equations and Lyapunov functions”. *arXiv:1903.05671*.

Smith, D. A., W. F. Ford, and A. Sidi. 1987. “Extrapolation methods for vector sequences”. *SIAM review*. 29(2): 199–233.

Solodov, M. V. and B. F. Svaiter. 1999a. “A hybrid approximate extragradient–proximal point algorithm using the enlargement of a maximal monotone operator”. *Set-Valued Analysis*. 7(4): 323–345.

Solodov, M. V. and B. F. Svaiter. 1999b. “A hybrid projection-proximal point algorithm”. *Journal of convex analysis*. 6(1): 59–70.

Solodov, M. V. and B. F. Svaiter. 2000. “Error bounds for proximal point subproblems and associated inexact proximal point algorithms”. *Mathematical Programming*. 88(2): 371–389.

Solodov, M. V. and B. F. Svaiter. 2001. “A unified framework for some inexact proximal point algorithms”. *Numerical functional analysis and optimization*. 22(7-8): 1013–1035.

Stiefel, E. 1952. “Methods of conjugate gradients for solving linear systems”. *Journal of Research of the National Bureau of Standards*. 49: 409–435.

Straeter, T. A. 1971. “On the extension of the davidon-broyden class of rank one, quasi-newton minimization methods to an infinite dimensional Hilbert space with applications to optimal control problems”. *Tech. rep.*

Sturm, J. F. 1999. “Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones”. *Optimization Methods and Software*. 11–12: 625–653.

Su, W., S. Boyd, and E. Candes. 2014. "A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights". In: *Advances in Neural Information Processing Systems (NIPS)*.

Süli, E. and D. F. Mayers. 2003. *An introduction to numerical analysis*. Cambridge university press.

Sun, B., J. George, and S. Kia. 2020. "High-Resolution Modeling of the Fastest First-Order Optimization Method for Strongly Convex Functions". In: *Proceedings of the 59th Conference on Decision and Control (CDC)*.

Sundararajan, A., B. Van Scoy, and L. Lessard. 2020. "Analysis and design of first-order distributed optimization algorithms over time-varying graphs". *IEEE Transactions on Control of Network Systems*. 7(4): 1597–1608.

Taylor, A. and F. Bach. 2019. "Stochastic first-order methods: non-asymptotic and computer-aided analyses via potential functions". In: *Proceedings of the 32nd Conference on Learning Theory (COLT)*.

Taylor, A. and Y. Drori. 2021. "An optimal gradient method for smooth strongly convex minimization". *arXiv:2101.09741*.

Taylor, A., B. Van Scoy, and L. Lessard. 2018a. "Lyapunov functions for first-order methods: Tight automated convergence guarantees". In: *Proceedings of the 35th International Conference on Machine Learning (ICML)*.

Taylor, A. B., J. M. Hendrickx, and F. Glineur. 2017a. "Exact worst-case performance of first-order methods for composite convex optimization". *SIAM Journal on Optimization*. 27(3): 1283–1313.

Taylor, A. B., J. M. Hendrickx, and F. Glineur. 2017b. "Performance estimation toolbox (PESTO): automated worst-case analysis of first-order optimization methods". In: *Proceedings of the 56th Conference on Decision and Control (CDC)*.

Taylor, A. B., J. M. Hendrickx, and F. Glineur. 2017c. "Smooth strongly convex interpolation and exact worst-case performance of first-order methods". *Mathematical Programming*. 161(1-2): 307–345.

Taylor, A. B., J. M. Hendrickx, and F. Glineur. 2018b. "Exact worst-case convergence rates of the proximal gradient method for composite convex minimization". *Journal of Optimization Theory and Applications*. 178(2): 455–476.

Teboulle, M. 2018. "A simplified view of first order methods for optimization". *Mathematical Programming*. 170(1): 67–96.

Toh, K.-C., M. J. Todd, and R. H. Tütüncü. 2012. "On the implementation and usage of SDPT3–a Matlab software package for semidefinite-quadratic-linear programming, version 4.0". In: *Handbook on semidefinite, conic and polynomial optimization*. Springer. 715–754.

Toth, A. and C. Kelley. 2015. "Convergence analysis for Anderson acceleration". *SIAM Journal on Numerical Analysis*. 53(2): 805–819.

Tseng, P. 2008. "On accelerated proximal gradient methods for convex-concave optimization". URL: http://www.mit.edu/~dimitrib/PTseng/papers.html.

Tseng, P. 2010. "Approximation accuracy, gradient methods, and error bound for structured convex optimization". *Mathematical Programming*. 125(2): 263–295.

Tyrtyshnikov, E. E. 1994. "How bad are Hankel matrices?" *Numerische Mathematik*. 67(2): 261–269.

Van Scoy, B., R. A. Freeman, and K. M. Lynch. 2017. "The fastest known globally convergent first-order method for minimizing strongly convex functions". *IEEE Control Systems Letters*. 2(1): 49–54.

Vandenberghe, L. and S. Boyd. 1999. "Applications of semidefinite programming". *Applied Numerical Mathematics*. 29(3): 283–299.

Villa, S., S. Salzo, L. Baldassarre, and A. Verri. 2013. "Accelerated and inexact forward-backward algorithms". *SIAM Journal on Optimization*. 23(3): 1607–1633.

Walker, H. F. and P. Ni. 2011. "Anderson acceleration for fixed-point iterations". *SIAM Journal on Numerical Analysis*. 49(4): 1715–1735.

Wibisono, A., A. C. Wilson, and M. I. Jordan. 2016. "A variational perspective on accelerated methods in optimization". In: *Proceedings of the National Academy of Sciences*.

Wilson, A. C., B. Recht, and M. I. Jordan. 2021. "A Lyapunov Analysis of Accelerated Methods in Optimization". *The Journal of Machine Learning Research (JMLR)*. 22(113): 1–34.

Wynn, P. 1956. "On a device for computing the $e_m(S_n)$ transformation". *Mathematical Tables and Other Aids to Computation*. 10(54): 91–96.

Xiao, L. 2010. "Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization". *The Journal of Machine Learning Research (JMLR)*. 11: 2543–2596.

Zhang, G., X. Bao, L. Lessard, and R. Grosse. 2021. "A unified analysis of first-order methods for smooth games via integral quadratic constraints". *The Journal of Machine Learning Research (JMLR)*. 22(103): 1–39.

Zhou, K., Q. Ding, F. Shang, J. Cheng, D. Li, and Z.-Q. Luo. 2019. "Direct acceleration of SAGA using sampled negative momentum". In: *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Zhou, K., F. Shang, and J. Cheng. 2018. "A Simple Stochastic Variance Reduced Algorithm with Fast Convergence Rates". In: *Proceedings of the 35th International Conference on Machine Learning (ICML)*.

Zhou, K., A. M.-C. So, and J. Cheng. 2020. "Boosting First-order Methods by Shifting Objective: New Schemes with Faster Worst Case Rates". In: *Advances in Neural Information Processing Systems (NeuRIPS)*.

Zhou, Z. and A. M.-C. So. 2017. "A unified approach to error bounds for structured convex optimization problems". *Mathematical Programming*: 1–40.

Zhou, Z., Q. Zhang, and A. M.-C. So. 2015. "l1, p-Norm Regularization: Error Bounds and Convergence Rate Analysis of First-Order Methods". In: *Proceedings of the 32nd International Conference on Machine Learning (ICML)*.