Lyapunov Arguments in Optimization

by

Ashia Wilson

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Statistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Michael Jordan, Co-chair
Professor Benjamin Recht, Co-chair
Professor Martin Wainwright
Professor Craig Evans

Spring 2018

Lyapunov Arguments in Optimization

Abstract

Lyapunov Arguments in
Optimization

by

Ashia Wilson

Doctor of Philosophy in Statistics

University of California, Berkeley

Professor Michael Jordan, Co-chair

Professor Benjamin Recht, Co-chair

Optimization is among the richest modeling languages in science. In statistics and machine learning, for instance, inference is typically posed as an optimization problem. While there are many algorithms designed to solve optimization problems, and a seemingly greater number of convergence proofs, essentially all proofs follow a classical approach from dynamical systems theory: they present a Lyapunov function and show it decreases. The primary goal of this thesis is to demonstrate that making the Lyapunov argument explicit greatly simplifies, clarifies, and to a certain extent, unifies, convergence theory for optimization.

The central contributions of this thesis are the following results: we

- present several variational principles whereby we obtain continuous-time dynamical systems useful for optimization;
- introduce Lyapunov functions for both the continuous-time dynamical systems and discrete-time algorithms and demonstrate how to move between these Lyapunov functions;
- utilize the Lyapunov framework as well as numerical analysis and integration techniques to obtain upper bounds for several novel discrete-time methods for optimization, a few of which have matching lower bounds.

For my family

Contents

List of Figures

List of Tables

Acknowledgments

Mom, Dad, Ayana and Jay, thank you for your unwavering love and support. You have been in the trenches with me and I certainly would not have made it to this point without you. You have my eternal gratitude and love. We did it!

To my advisors, Mike and Ben, thank you, thank you, thank you for providing me with invaluable encouragement, guidance, and support throughout the course of my PhD. Other academic mentors to whom I owe special thanks are Cynthia Rudin, Eric Tchetgen Tchetgen, Pamela Abshire, and Michael Brenner. Thank you for your kindness.

Thank you to my collaborators, Andre, Nick, Tamara, Becca, Shivaram, Mitchell, Stephen, Micheal B., Alex and Nati Srebro. I am so fortunate to have worked with and learned from such brilliant and kind people. I must single out Andre – you are a soulmate – and Nick – thank you for being such a patient and loving friend. Becca, girl, so grateful for your energy.

I am honored and grateful to have many truly amazing and supportive relationships in my life. While there are far too many to provide an exhaustive list, I would be remiss if I did not mention some key friends. Henry, Brett, Velencia, Robert, Meron, Kene, Christine, Marianna, Nina, Dawn, Po-Ling, and Nick A. Velencia and Meron, you are sisters. Thank you for showing up, and showing up, and showing up. Henry (and Lucy), Brett, and Robert, thank you for always being there. And to some new close friends, Jee, Elliot, Rocky, Jamal, Mike. You inspire me and grow me. I hope to know you for a long time.

I am also very grateful for the The Berkeley Chancellor's Postdoctoral Fellowship Program and the National Science Foundation for providing me with financial support.

Chapter 1

Introduction

The ubiquity of optimization problems in science and engineering has led to great interest in the design of algorithms meant to solve them. This thesis discusses the connection between continuous-time dynamical systems and discrete-time algorithms for machine learning and optimization. Examples of the algorithms we discuss include the stochastic, proximal, coordinate, and deterministic variants of gradient descent, mirror descent, dual averaging, accelerated gradient descent, the conditional gradient method, dual averaging with momentum, and adaptive gradient methods. For each algorithm, we show that a single, simple Lyapunov argument proves convergence. In this chapter, we present background material.

1.1 Preliminary Concepts

We begin by defining what an optimizatin problem is, what an algorithm is, and what it means to have an upper-bound for an algorithm. We also define continuous-time dynamical systems and discuss Lyapunov's method. We end by describing our strategy for moving between the Lyapunov arguments presented in continuous time and in discrete time; this will provide a framework for obtaining upper bounds on the rate of convergence for most algorithms in optimization.

1.1.1 Optimization

The field of optimization studies the following problem,

$$\min_{x \in \mathcal{X}} f(x). \tag{1.1}$$

Here, x is the decision variable, $\mathcal{X}$ is the set of possible decisions, and $f : \mathcal{X} \to \mathbb{R}$ is the objective function. Throughout, we refer to $x^* \in \arg\min_{x \in \mathcal{X}} f(x)$ as a solution to (1.1). Most decision and inference problems in engineering and science are modeled as (1.1). In Appendix A, we list several motivating examples.

1.1.2 Algorithms and Upper Bounds

An optimization algorithm is a recipe for generating a sequence of points $(x_s)_{s=0}^k$ to solve problem (1.1). To generate the next point in the sequence, the algorithm uses the local information it receives from an *oracle* along with all the previous information it has received. An oracle is a black-box function that provides the algorithm with information about $(f, \mathcal{X})$ at a point x. As a specific example, if f is differentiable and $\mathcal{X} = \mathbb{R}^d$, a gradient oracle provides the algorithm with the function value and the gradient $(f(x), \nabla f(x))$ at any queried point x; algorithms which have access to this oracle function use the information $(x_s)_{s=0}^k$, $(f(x_s))_{s=0}^{k-1}$, $(\nabla f(x_s))_{s=0}^{k-1}$, and the pair $(f(x), \nabla f(x))$ computed at any point x, to construct the next point x_{k+1}.

Suppose two algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ are proposed to solve an instance of problem (1.1). Which algorithm should we choose? Is there another algorithm $\mathcal{A}_3$ which finds x^* faster than $\mathcal{A}_1$ and $\mathcal{A}_2$? The goal of complexity theory for optimization is to address these questions. Ideally, we could partition the space of problems $(f, \mathcal{X})$ according to which algorithms are fastest for solving them (or vice versa). In addition, we might search for a framework that takes into account potential computational constraints. Unfortunately, given the size of the space of possible problems, creating such a comprehensive framework is probably impossible.

Instead, the standard adopted widely in the theory community is to treat classes of problems, and evaluate algorithms according to the *minimum* number of oracle queries required to produce an *approximate solution* for any problem instance in the class. An approximate solution is an iterate x_k such that $f(x_k) - f(x^*) \leq \epsilon$, $d(x_k, x^*) \leq \epsilon$, or $d^*(\nabla f(x_k), \nabla f(x^*)) \leq \epsilon$ for some distance measure $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ or $d^* : \mathcal{X}^* \times \mathcal{X}^* \to \mathbb{R}^+$ and error threshold $\epsilon > 0$.

The minimum number of oracle queries required to find an approximate solution for any function in a prespecified class of functions is called a *lower bound* for the oracle. For example, consider $f \in \mathcal{F}$, where $\mathcal{F}$ is the class of convex functions with smooth gradients over $\mathbb{R}^d$. Any algorithm which has access to the gradient oracle for this class of functions requires a minimum $\Omega(1/\sqrt{\epsilon})$ queries to find a solution such that $f(x_k) - f(x^*) \leq \epsilon$.

Let x_k be the output of the algorithm on the k-th iteration. An *upper bound* for an algorithm is a sequence $\epsilon(k)$, such that $f(x_k) - f(x^*) \leq \epsilon(k)$, $d(x_k, x^*) \leq \epsilon(k)$, or $d^*(\nabla f(x_k), \nabla f(x^*)) \leq \epsilon(k)$. Generally speaking, upper bounds quantify how fast a solution to (1.1) is being found by the algorithm. An algorithm $\mathcal{A}$ is called *provably optimal* if there is an upper bound which matches the lower bound for the oracle function it has access to. While this thesis does not discuss lower bounds or techniques for deriving them, we mention when it has been established in the literature that an algorithm is provably optimal.

1.1.3 Role of Convergence Theorems

Our goal in discussing how to obtain upper bounds for algorithms is to make explicit the connection between continuous-time dynamical systems and discrete-time algorithms for optimization as well as the Lyapunov arguments used to analyze both. The observation that

Lyapunov arguments are important to convergence theory in optimization is not new; in his book, *Introduction to Optimization*, for instance, Polyak makes this specific point [54]. A main contribution of this thesis is to discuss the Lyapunov analysis of several methods that are not covered in Polyak's early book, and to make the connection between continuous-time dynamical systems and discrete-time dynamical systems more concrete.

In the same book, Polyak encourages his reader to proceed with caution when studying convergence theory for optimization. From the perspective of practitioners, he acknowledges, the conditions under which the bound can be obtained are often hard to verify, unknown, or frequently violated. Furthermore, it is unclear whether worst-case performance over a function class is the criterion by which we should measure the performance of algorithms.

Nevertheless, convergence proofs provide useful information about the algorithm. Convergence guarantees determine a class of problems for which one can count on the applicability of the algorithm, as well as provide information on the qualitative behavior of convergence: whether we should expect convergence for any initial approximation or only for a sufficiently good one, and in what sense we should expect the converges to happen (the function converges, or the argument, non-asymptotic vs asymptotic, and so on).

1.1.4 Dynamical Systems

A dynamical system is a time varying vector field $v : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$. From an initial point X_0, a dynamical system generates a curve, called a trajectory, via the equation

$$X_t = X_0 + \int_0^t v_s(X_s)ds, \tag{1.2}$$

where we adopt the shorthand $v_t(x) := v(t, x)$. We can interpret $v_t(x)$ as a velocity at position x,

$$\frac{d}{dt}X_t = v_t(X_t). \tag{1.3}$$

In this thesis, we study how dynamical systems defined by ordinary differential equations (1.3) can be used to generate discrete sequences of points. Notably, most algorithms in optimization are obtained from applying either the forward or backward Euler method to (1.3). We provide a short explanation of these two techniques.

Suppose we are given a dynamical system (1.3) and a starting position $X_t \in \mathbb{R}^d$. The goal is to use X_t and its velocity $v_t(X_t)$ to tell us where to move next in time. To do so, we adopt a scaling of time, $\delta > 0$, (i.e. our notion of next) and approximate $X_{t+\delta}$ from X_t. Using the integral curve formulation (1.2), we have,

$$X_{t+\delta} - X_t = \int_t^{t+\delta} v_s(X_s)ds. \tag{1.4}$$

We can form a discrete sequence of points as follows. For any initial point X_t, approximation of the integral (1.4) by its upper-limit, $v_{t+\delta}(X_{t+\delta})\delta$, defines an operator called the

Backward-Euler (BE) method. In particular, if we write $x_{k+1} := X_{\delta(k+1)} = X_{t+\delta}$ and $x_k := X_{\delta k} = X_t$ and make the same identifications for the vector field, $v_k(x_k) := v_k(X_{\delta k}) = v_t(X_t)$ and $v_{k+1}(x_{k+1}) := v_{k+1}(X_{\delta(k+1)}) = v_{t+\delta}(X_{t+\delta})$, we can write the BE method as,

$$\frac{x_{k+1} - x_k}{\delta} = v_{k+1}(x_{k+1}). \tag{1.5}$$

Approximation of the integral by its lower-limit, $v_t(X_t)\delta$, defines another operator called the *Forward-Euler* (FE) method. Using the same identiifcations, we can write the FE method as,

$$\frac{x_{k+1} - x_k}{\delta} = v_k(x_k). \tag{1.6}$$

Both the BE method (1.5),

$$x_{k+1} = (\mathsf{I} - \delta v_{k+1})^{-1}(x_k) := \mathcal{A}^{\mathrm{BE}}_{\delta,v}(x_k),$$

and FE method (1.6),

$$x_{k+1} = (\mathsf{I} + \delta v_k)(x_k) := \mathcal{A}^{\mathrm{FE}}_{\delta,v}(x_k),$$

applied to dynamics (1.3) form discrete-time dynamical systems parameterized by the vector field v and discretization scaling δ. These discrete-time dynamical systems are equivalent to algorithms for oracle functions that allow the algorithm to compute $\mathcal{A}^{\mathrm{BE}}_{\delta,v}$ or $\mathcal{A}^{\mathrm{FE}}_{\delta,v}$ evaluated at any point $x \in \mathcal{X}$ each time it is queried. As an example, suppose $v_k \equiv \nabla f$. The FE operator, is a popular algorithm called gradient descent, $x_{k+1} = x_k - \delta \nabla f(x_k)$, and the BE operator is another algorithm called the proximal method. We elaborate on this example, as well as provide many more examples in Chapter 2.

1.1.5 Lyapunov's Method

A popular way to describe dynamical systems is via *conserved* and *dissipated* quantities. The general technique prescribes identifying a quantity $\mathcal{E} : \mathcal{X} \to \mathbb{R}$ which is either *constant* (conserved),

$$\frac{d}{dt}\mathcal{E}(X_t) = \langle \nabla \mathcal{E}(X_t), v_t(X_t) \rangle = 0, \tag{1.7a}$$

decreasing (dissipated),

$$\frac{d}{dt}\mathcal{E}(X_t) = \langle \nabla \mathcal{E}(X_t), v_t(X_t) \rangle \leq 0, \tag{1.7b}$$

or *strictly decreasing*,

$$\frac{d}{dt}\mathcal{E}(X_t) = \langle \nabla \mathcal{E}(X_t), v_t(X_t) \rangle < 0, \tag{1.7c}$$

along the trajectories of the dynamical system (1.3).

This thesis is primarily concerned with dissipated quantities for dynamical systems (1.7b) that have explicit dependencies on time, which we refer to as *Lyapunov functions*. The idea of providing the trajectories of dynamical systems with qualitative descriptions was formulated by Lyapunov in his fundamental work [36]; there exist many textbooks and monographs expanding on this idea (see [27, 19] for example).

From Lyapunov Functions to Convergence Theorems We demonstrte more explicitly how Lyapunov functions will be used to obtain upper bounds for most algorithms in optimization. Suppose we have generated a trajectory of the dynamical system X_t from an arbritrary starting position X_0. To provide bounds for the rate at which $\mathcal{E}_1 = f(X_t) - f(x^*)$, $\mathcal{E}_2 = d(X_t, x^*)$, or $\mathcal{E}_3 = d^*(\nabla f(X_t), \nabla f(x^*))$, converge to zero, we will consider $\mathcal{E}_1$, $\mathcal{E}_2$, and/or $\mathcal{E}_3$, as well as combinations of them, scaled by some function of time. For example, if we show the time-dependent function

$$\mathcal{E}_t = e^{\beta_t}(f(X_t) - f(x^*)), \tag{1.8}$$

is a Lyapunov function, where $\beta_t : \mathbb{R} \to \mathbb{R}$ is an arbitrary smooth, continuously differentiable, increasing function of time, $\frac{d}{dt}\beta_t > 0$, then we will be able to conclude a non-asymptotic rate of convergence. Specifically, if (1.8) is decreasing, then $\frac{d}{dt}\mathcal{E}_t \leq 0$ (i.e (1.7b) holds); by integrating we can conclude the property, $\mathcal{E}_t \leq \mathcal{E}_0$, which will allow us to obtain the upper bound,

$$f(X_t) - f(x^*) \leq \frac{\mathcal{E}_0}{e^{\beta_t}} = \frac{e^{\beta_0}(f(X_0) - f(x^*))}{e^{\beta_t}},$$

and subsequently, an $O(e^{-\beta_t})$ convergence rate.

In discrete time, we start by mapping the dynamical system to an algorithm, which generates a discrete sequence of points from any given starting point x_0. To provide bounds for the rate at which $E_1 = f(x_k) - f(x^*)$, $E_2 = d(x_k, x^*)$, and/or $E_3 = d^*(\nabla f(x_k), \nabla f(x^*))$, converge to zero, the strategy will be same. Following the above example, we consider the Lyapunov function,

$$E_k = A_k(f(x_k) - f(x^*)),$$

where $A_k : \mathbb{R} \to \mathbb{R}_+$ is an increasing sequence in k. If we make the identifications $e^{\beta_{t+\delta}} = A_{k+1}$ and $e^{\beta_t} = A_k$, then the requirement $\frac{d}{dt}\beta_t > 0$ translates to the requirement $\frac{d}{dt}\beta_t = \frac{d}{dt}e^{\beta_t}/e^{\beta_t} \approx (A_{k+1} - A_k)/\delta A_k > 0$. This, of course, is based on adopting an exponential scaling of time e^{β_t}. If, instead, we choose to scale time linearly, $\mathcal{E}_t = \tau_t(f(X_t) - f(x^*))$, so that $\tau_t = A_k$ and $\tau_{t+\delta} = A_{k+1}$, then the requirment $\frac{d}{dt}\tau_t > 0$ translates to the requirement $\frac{d}{dt}\tau_t \approx (A_{k+1} - A_k)/\delta > 0$. These two ways of scaling time appear throughout this framework. For either approximation, we check whether the Lyapunov property, $\frac{E_{k+1} - E_k}{\delta} \leq 0$, can be shown for various discretizations of the dynamical system. If so, by summing we can show, $E_k \leq E_0$, which will allow us to obtain the upper bound,

$$f(x_k) - f(x^*) \leq \frac{E_0}{A_k} = \frac{A_0(f(X_0) - f(x^*))}{A_k},$$

for the algorithm, and subsequently, a matching $O(1/A_k)$ convergence rate.

We provide several specific examples of this technique in Chapters 2 and 3.

1.2 Goals and Organization

The primary contribution of this thesis is to present and discuss Lyapunov arguments commonly used in optimization. The Lyapunov framework we present demonstrates the centrality of dynamical systems to the field of optimization and machine learning. We organize this thesis as follows:

- Chapter 2 summarizes several families of ordinary differential equations used in optimization. Section 2.1 focuses on algorithms in optimization that discretize first-order differential equations (i.e. ODEs with one time derivative). This includes gradient descent, mirror descent, subgradient methods, dual averaging, and the condiitional gradient algorithms. For all these algorithms, we demonstrate how to move between the Lyapunov arguments presented in continuous and discrete-time. This material will mostly be presented in tables with more complete descriptions provided in the Appendices.

- Section 2.2 focuses on algorithms in opimization that discretize second-order differential equations (i.e. ODEs with two time derivatives). This includes accelerated gradient descent, the accelerated proximal gradient method and the quasi-monotone subgradient methods. Techniques for obtaining upper-bounds for these algorithms have been famously considered esoteric. We demonstrate how many of these techniques, including the technique of *estimate sequences*, are equivalent to a Lyapunov argument. In addition, we introduce two Lagrangian/Hamiltonian functionals, we call Bregman Lagrangians/Hamiltonians, which generate two large classes of accelerated methods in continuous time. We then provide a systematic methodology for converting the continuous-time dynamical systems obtained from these variational principles to discrete-time algorithms with matching convergence rates.

- Chapter 3 extends the Lyapunov framework to stochastic differential equations and stochastic algorithms. This includes stochastic gradient descent, stochastic mirror descent, stochastic dual averaging, stochastic accelerated gradient descent. We also present an analogous description for coordinate-based methods. In particular, we show the same Lyapunov functions presented in the previous chapter provide upper bounds for the rate at which these algorithms find a solution to (1.1) in expectation.

How to read this thesis A good strategy for gleaming the content of this thesis is to review the tables at the beginning of each subsection and to skim the summary section presented at the end of both chapters.

Chapter 2

Deterministic Dynamical Systems

In this chapter, we summarize several families of dynamics (1.3) which can be said to find a solution to (1.1). For each dynamic, we exhibit a Lyapunov function which will ensure convergence to a stationary point that is either a solution to (1.1) or a critical point of the objective function. We also show how to move between the continuous and discrete-time analyses using two standard discretization methods.

2.1 Lyapunov Analysis of First-Order Dynamics

2.1.1 Gradient Descent Dynamic

Let $\mathcal{X} = \mathbb{R}^d$. The dynamic that gives rise to gradient descent can be analyzed in (at least) four different settings:

1. f is differentiable, but not necessarily convex;
2. f is convex, so that $D_f(x, y) \geq 0 \ \forall x, y \in \mathcal{X}$ (see (A.2) for definition) ;
3. f satisfies the Polyak-Löjasiewicz (PL) condition with parameter μ, so that

$$-\frac{1}{2}\|\nabla f(x)\|^2 \leq -\mu(f(x) - f(x^*)), \ \forall x \in \mathcal{X}. \tag{2.1}$$

4. f is μ-strongly convex, so that $D_f(y, x) \geq \frac{\mu}{2}\|y - x\|^2$.

We summarize the Lyapunov functions in Table 2.1 and provide a description of the first and second bullet points in the main text. The rest of the results can be found in Appendix B.1.

The gradient descent dynamic (GF),

$$\frac{d}{dt}X_t = \arg\min_{v \in \mathbb{R}^d} \left\{ \langle \nabla f(x), v \rangle + \frac{1}{2}\|v\|^2 \right\} = -\nabla f(X_t), \tag{2.2}$$

Gradient Flow:	$\dot{X}_t = -\nabla f(X_t)$	
Function Class	Lyapunov Function	Convergence Rate
Differentiable	$\mathcal{E}_t = f(X_t) - f(x^*)$	$\min_{0\le s\le t}\|\nabla f(X_s)\| \le O(1/t^{\frac{1}{2}})$
Convex	$\mathcal{E}_t = \frac{1}{2}\|x^* - X_t\|^2 + t(f(X_t) - f(x^*))$	$f(X_t) - f(x^*) \le O(1/t)$
PL Condition w.p μ	$\mathcal{E}_t = e^{2\mu t}(f(X_t) - f(x^*))$	$f(X_t) - f(x^*) \le O(e^{-2\mu t})$
μ*-Strong Convexity*	$\mathcal{E}_t = e^{\mu t}\frac{1}{2}\|x^* - X_t\|^2$	$\frac{1}{2}\|x^* - X_t\|^2 \le O(e^{-\mu t})$
f is L-smooth	$\mathcal{E}_t = e^{\frac{2\mu L}{\mu+L}t}\frac{1}{2}\|x^* - X_t\|^2$	$\frac{1}{2}\|x^* - X_t\|^2 \le O(e^{-\frac{2\mu L}{\mu+L}t})$
		$f(X_t) - f(x^*) \le O(\frac{L}{2}e^{-\frac{2\mu L}{\mu+L}t})$
Gradient Descent:	$\frac{x_{k+1}-x_k}{\delta} = -\nabla f(x_k)$	
Function Class	Lyapunov Function	Convergence Rate
Differentiable f is $(1/\delta)$-smooth	$E_k = f(x_k) - f(x^*)$	$\min_{0\le s\le k}\frac{1}{2}\|\nabla f(x_s)\| \le O(1/(\delta k)^{\frac{1}{2}})$
Convex f is $(1/\delta)$-smooth	$E_k = \frac{1}{2}\|x^* - x_k\|^2 + \delta k(f(x_k) - f(x^*))$	$f(x_k) - f(x^*) \le O(1/\delta k)$
PL Condition w.p μ f is $(1/\delta)$-smooth	$E_k = (1-\mu\delta)^{-k}(f(x_k) - f(x^*))$	$f(x_k) - f(x^*) \le O(e^{-\mu\delta k})$
μ*-Strong Convexity* f is $(1/\delta)$-smooth	$E_k = (1-\mu\delta)^{-k}\frac{1}{2}\|x^* - x_k\|^2$	$\frac{1}{2}\|x^* - x_k\|^2 \le O(e^{-\mu\delta k})$
f is $(L = \frac{2-\mu\delta}{\delta})$-smooth	$E_k = \left(1 - \frac{2\mu L}{\mu+L}\delta\right)^{-k}\frac{1}{2}\|x^* - x_k\|^2$	$\frac{1}{2}\|x^* - x_k\|^2 \le O(e^{-\frac{2\mu L}{\mu+L}\delta k})$
		$f(x_k) - f(x^*) \le O(\frac{L}{2}e^{-\frac{2\mu L}{\mu+L}\delta k})$
Proximal Method:	$\frac{x_{k+1}-x_k}{\delta} = -\nabla f(x_{k+1})$	
Function Class	Lyapunov Function	Convergence Rate
Differentiable $\delta > 0$	$E_k = f(x_k) - f(x^*)$	$\min_{0\le s\le t}\frac{1}{2}\|\nabla f(x_s)\| \le O(1/(\delta k)^{\frac{1}{2}})$
Convex $\delta > 0$	$E_k = \frac{1}{2}\|x^* - x_k\|^2 + \delta k(f(x_k) - f(x^*))$	$f(x_k) - f(x^*) \le O(1/\delta k)$
PL Condition w.p μ $\delta > 0$	$E_k = (1+\mu\delta)^{k}(f(x_k) - f(x^*))$	$f(x_k) - f(x^*) \le O(e^{-\mu\delta k})$
μ*-Strong Convexity* $\delta > 0$	$E_k = (1+\mu\delta)^{k}\frac{1}{2}\|x^* - x_k\|^2$	$\frac{1}{2}\|x^* - x_k\|^2 \le O(e^{-\mu\delta k})$
f is L-smooth	$E_k = \left(1 + \frac{2\mu L}{\mu+L}\delta\right)^{k}\frac{1}{2}\|x^* - x_k\|^2$	$\frac{1}{2}\|x^* - x_k\|^2 \le O(e^{-\frac{2\mu L}{\mu+L}\delta k})$
		$f(x_k) - f(x^*) \le O(\frac{L}{2}e^{-\frac{2\mu L}{\mu+L}\delta k})$

Table 2.1: Lyapunov functions for gradient flow (GF), gradient descent (GD), and the proximal method (PM); with discrete-time identification $t = \delta k$, the results in continuous time and discrete time match up to a constant factor of 2.

is a steepest descent flow. For any initial starting point X_0, GF moves with velocity $v_t(X_t) =$

$-\nabla f(X_t)$, and stops only at a critical point of the function ($\frac{d}{dt}X_t = 0$ if and only if $\nabla f(X_t) = 0$). Gradient descent (GD),

$$\frac{x_{k+1} - x_k}{\delta} = \arg\min_{v \in \mathbb{R}^d} \left\{ \langle \nabla f(x), v \rangle + \frac{1}{2}\|v\|^2 \right\} = -\nabla f(x_k), \tag{2.3}$$

is the result of applying the forward-Euler method (1.6) to GF (2.2). GD similarly adds to its current position x, the gradient $\nabla f(x)$ computed at x, and stops only at a critical point of the function ($\frac{x_{k+1}-x_k}{\delta} = 0$ if and only if $\nabla f(x_k) = 0$). The backward-Euler method (1.5) applied to (2.2),

$$\frac{x_{k+1} - x_k}{\delta} = -\nabla f(x_{k+1}), \tag{2.4}$$

is called the proximal method (PM). It is a stationary point of the following optimization problem,

$$x_{k+1} \in \arg\min_{x \in \mathcal{X}} \left\{ f(x) + \frac{1}{2\delta}\|x - x_k\|^2 \right\} := \mathrm{Prox}_{\delta f}(x_k). \tag{2.5}$$

Given the nature of the update (2.5), the PM is used primarily when f is easy to optimize over (such as when $f(x) = \|x\|_1$ is the ℓ_1 norm). See [53] for an excellent monograph on the proximal methods and the various ways to interpret them. Lyapunov analyses for all these methods follow a similar structure.

2.1.1.1 Nonconvex Differentiable Functions

The optimality gap,

$$\mathcal{E}_t = f(X_t) - f(x^*)$$

is a Lyapunov function for (2.2). We check,

$$\frac{d}{dt}\mathcal{E}_t = \frac{d}{dt}f(X_t) = \langle \nabla f(X_t), \dot{X}_t \rangle \overset{(2.2)}{=} -\|\nabla f(X_t)\|^2. \tag{2.6}$$

Here, the first equality follows because x^* is constant with respect to time, and the second equality uses the chain rule. By rearranging and integrating (2.6), $t \min_{0\le s\le t} \|\nabla f(X_s)\|^2 \le \int_0^t \|\nabla f(X_s)\|^2 ds \le \mathcal{E}_0 - \mathcal{E}_t \le \mathcal{E}_0$, we conclude a $O(1/t)$ convergence to a critical point of the function,

$$\min_{0\le s\le t} \|\nabla f(X_s)\|^2 \le \frac{f(X_0) - f(x^*)}{t}.$$

Given the description above, this result is intuitive – if we go down hill, we will only stop at a critical point of the function, i.e. point x where $\nabla f(x) = 0$. This corresponds to saddle points and local/global minimizers of the function, if they exist. Similar statements can be made for gradient descent and the proximal method.

Gradient Descent As long as the function is L-smooth (A.13) , where $0 \leq \delta < 2/L$, the optimality gap,

$$E_k = f(x_k) - f(x^*), \tag{2.7}$$

remains a Lyapunov function for GD. We check,

$$\frac{E_{k+1} - E_k}{\delta} = \frac{f(x_{k+1}) - f(x_k)}{\delta} \leq \frac{2 - \delta L}{2} \left\langle \nabla f(x_k), \frac{x_{k+1} - x_k}{\delta} \right\rangle \overset{(2.3)}{=} -\frac{2 - \delta L}{2} \|\nabla f(x_k)\|^2. \tag{2.8}$$

Here, the inequality follows from the smoothness condition (A.13). Take $L = 1/\delta$. We can similarly rearrange this statement to conclude, $\delta k \min_{0 \leq s \leq t} \frac{1}{2}\|\nabla f(x_s)\|^2 \leq \delta \sum_{s=0}^{k} \frac{1}{2}\|\nabla f(x_s)\|^2 \leq E_0 - E_t \leq E_0$. Therefore, as long as f is a $(1/\delta)$-smooth function, we can guarantee $O(1/\delta k)$ convergence to a stationary point,

$$\min_{0 \leq s \leq k} \frac{1}{2}\|\nabla f(x_s)\|^2 \leq \frac{f(x_0) - f(x^*)}{\delta k}.$$

Proximal Method The discrete optimality gap (2.7) is a Lyapunov function for (2.5) as well. We check,

$$\frac{E_{k+1} - E_k}{\delta} = \frac{f(x_{k+1}) - f(x_k)}{\delta} \leq \frac{1}{2} \left\langle \nabla f(x_{k+1}), \frac{x_{k+1} - x_k}{\delta} \right\rangle \overset{(2.4)}{=} -\frac{1}{2}\|\nabla f(x_{k+1})\|^2. \tag{2.9}$$

The inequality follows from the optimality condition (2.5), which implies $f(x_{k+1}) + \frac{1}{2\delta}\|x_{k+1} - x_k\|^2 \leq f(x_k) + \frac{1}{2\delta}\|x_k - x_k\|^2$. By rearranging and summing, we obtain the analogous discrete-time statement,

$$\min_{0 \leq s \leq k} \frac{1}{2}\|\nabla f(x_s)\|^2 \leq \frac{f(x_0) - f(x^*)}{\delta k}.$$

When moving between upper bounds in continuous and discrete time, we lose a factor of two and an additional assumption (smoothness for GD) is needed.

2.1.2 Convex Functions

When the objective function is convex, the Lyapunov function,

$$\mathcal{E}_t = \frac{1}{2}\|x^* - X_t\|^2 + t(f(X_t) - f(x^*)),$$

allows us to conclude convergence to the minimizer. We check,

$$\begin{aligned} \frac{d}{dt}\mathcal{E}_t &= -\left\langle \frac{d}{dt}X_t, x^* - X_t \right\rangle + f(X_t) - f(x^*) + t\frac{d}{dt}f(X_t) \\ &\overset{(2.2)}{=} -D_f(x^*, X_t) - t\|\nabla f(X_t)\|^2 \leq 0. \end{aligned}$$

Here, the inequality follows from convexity of f, which ensures the Bregman divergence (A.2) is non-negative. By integrating, we obtain the statement $\mathcal{E}_t - \mathcal{E}_0 = \int_0^t \frac{d}{ds}\mathcal{E}_s ds \leq 0$, from which we can conclude at $O(1/t)$ convergence rate of the function value,

$$f(X_t) - f(x^*) \leq \frac{\mathcal{E}_0}{t}.$$

Similar statements can be made about GF (2.2) and the PM (2.4).

Gradient Descent For GD, as long as the function is L-smooth, where $\delta \leq 1/L$, the following function,

$$E_k = \frac{1}{2}\|x^* - x_k\|^2 + \delta k(f(x_k) - f(x^*)), \tag{2.10}$$

is a Lyapunov function. For simplicity, take $\delta = 1/L$. We check,

$$\begin{aligned}\frac{E_{k+1} - E_k}{\delta} &= -\left\langle \frac{x_{k+1} - x_k}{\delta}, x^* - x_k \right\rangle + f(x_k) - f(x^*) + \delta k \frac{f(x_{k+1}) - f(x_k)}{\delta} + \varepsilon_k^1 \\ &\overset{\substack{(2.3)\\(2.8)}}{\leq} -D_f(x^*, x_k) - \frac{\delta k}{2}\|\nabla f(x_k)\|^2 + \varepsilon_k^2 \leq 0,\end{aligned}$$

where $\varepsilon_k^1 = f(x_{k+1}) - f(x_k) - \langle \frac{x_{k+1}-x_k}{\delta}, x_k - x_{k+1}\rangle - \frac{\delta}{2}\|\frac{x_{k+1}-x_k}{\delta}\|^2$ and $\varepsilon_k^2 = D_f(x_{k+1}, x_k) - \frac{\delta}{2}\|\nabla f(x_k)\|^2 \leq -(\frac{\delta}{2} - \frac{\delta^2 L}{2})\|\nabla f(x_k)\|^2 \leq 0$; the upper bound on the error follows from the smoothness assumption and the identification $\delta = 1/L$. The first inequality plugs in the algorithm (2.3) and the descent property (2.8). By summing, we obtain the statement $E_k - E_0 = \sum_{i=1}^k \frac{E_{i+1}-E_i}{\delta}\delta \leq 0$, from which we can conclude a $O(1/\delta k)$ convergence rate of the function value,

$$f(x_k) - f(x^*) \leq \frac{E_0}{\delta k}.$$

Proximal Method The function (2.10) is a Lyapunov function for the PM algorithm as well. We check,

$$\begin{aligned}\frac{E_{k+1} - E_k}{\delta} &= -\left\langle \frac{x_{k+1} - x_k}{\delta}, x^* - x_{k+1} \right\rangle + f(x_{k+1}) - f(x^*) + \delta(k+1)\frac{f(x_{k+1}) - f(x_k)}{\delta} + \varepsilon_k^1 \\ &\overset{(2.4)}{\leq} -D_f(x^*, x_{k+1}) - \frac{\delta(k+1)}{2}\|\nabla f(x_{k+1})\|^2 \leq 0,\end{aligned}$$

where $\varepsilon_k^1 = -\frac{\delta}{2}\|\frac{x_{k+1}-x_k}{\delta}\|^2$. The first inequality uses (2.9) and the last inequality follows from the convexity of f. By summing, we obtain the statement $E_k - E_0 = \sum_{i=1}^k \frac{E_{i+1}-E_i}{\delta}\delta \leq 0$, from which we can conclude a $O(1/\delta k)$ convergence rate of the function value,

$$f(x_k) - f(x^*) \leq \frac{E_0}{\delta k}.$$

2.1.3 Mirror Descent Dynamic

We analyze the dynamic that gives rise to mirror descent/natural gradient descent in four different settings:

1. f is differentiable, but not necessarily convex;
2. f is convex, so that $D_f(x, y) \geq 0 \ \forall x, y \in \mathcal{X}$;
3. f satisfies the Polyak-Löjasiewicz (PL) condition with parameter μ, so that
$$-\|\nabla f(x)\|_{x^*}^2 \leq -2\mu(f(x) - f(x^*)), \forall x \in \mathcal{X}. \tag{2.11}$$
Here, $\|v\|_{x^*} = \langle v, \nabla^2 h(x)^{-1} v\rangle$. When $h = \frac{1}{2}\|x\|^2$, this is equivalent to condition (2.1)
4. f is μ-strongly convex with respect to a strictly convex function h (A.7), so that $D_f(y, x) \geq \mu D_h(y, x) \ \forall x, y \in \mathcal{X}$.

We summarize the Lyapunov functions presented in this subsection in Table 2.2. The mirror descent dynamic is a natural generalization of the steepest descent dynamic to smooth manifolds $\tilde{\mathcal{X}}$, where the metric on $\tilde{\mathcal{X}}$ is given by the Hessian of a strictly convex function $h : \tilde{\mathcal{X}} \to \mathbb{R}$, and the objective function $f : \mathcal{X} \to \mathbb{R}$, is defined on $\mathcal{X} \subseteq \text{cl}(\tilde{\mathcal{X}})$, where $\mathcal{X} \cap \tilde{\mathcal{X}} \neq \emptyset$. For the remainder of this subsection, we will take $\mathcal{X} = \tilde{\mathcal{X}} = \mathbb{R}^d$ and provide a description of the second bullet point only; a presenentation of the more general setting as well as the other bullet points are described in Appendix B.2.

2.1.3.1 Convex Functions

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a Lipschitz, continuously differentiable convex function. The mirror descent dynamic (MF),
$$\frac{d}{dt}\nabla h(X_t) = -\nabla f(X_t), \tag{2.12}$$
is a steepest descent flow on with respect to the metric $\|v\|_x^2$ (see definition (B.9) for more details). Furthermore, the function
$$\mathcal{E}_t = D_h(x^*, X_t) + \int_0^t (f(X_s) - f(x^*))ds, \tag{2.13}$$
is a Lypaunov function for MF (2.12). We check,
$$\frac{d}{dt}\mathcal{E}_t = -\left\langle \frac{d}{dt}\nabla h(X_t), x^* - X_t \right\rangle + f(X_t) - f(x^*) \overset{(2.12)}{=} -D_f(x^*, X_t). \tag{2.14}$$

Mirror Descent Dynamic:	$\frac{d}{dt}\nabla h(X_t) = -\nabla f(X_t)$	
Function Class	Lyapunov Function	Convergence Rate
Convex	$\mathcal{E}_t = D_h(x^*, X_t) + \int_0^t f(X_s) - f(x^*)ds$	$f(\hat{X}_t) - f(x^*) \leq O(1/t)$
	$\mathcal{E}_t = D_h(x^*, X_t) + t((X_s) - f(x^*))$	$f(X_t) - f(x^*) \leq O(1/t)$
μ-Strong Convexity	$\mathcal{E}_t = e^{\mu t} D_h(x^*, X_t)$	$D_h(x^*, X_t) \leq O(e^{-\mu t})$
Mirror Descent:	$\frac{\nabla h(x_{k+1}) - \nabla h(x_k)}{\delta} = -\nabla f(x_k)$	
Function Class	Lyapunov Function	Convergence Rate
Convex f is $(1/\delta)$-smooth	$E_k = D_h(x^*, x_k) + \sum_{s=0}^{k}(f(x_s) - f(x^*))\delta$	$f(\hat{x}_k) - f(x^*) \leq O(1/\delta k)$
μ-Strong Convexity f is $(1/\delta)$-smooth	$E_k = (1 - \mu\delta)^{-k} D_h(x^*, x_k)$	$D_h(x^*, x_k) \leq O(e^{-\mu\delta k})$
Breg Prox Minimization:	$\frac{\nabla h(x_{k+1}) - \nabla h(x_k)}{\delta} = -\nabla f(x_{k+1})$	
Function Class	Lyapunov Function	Convergence Rate
Convex $\delta > 0$	$E_k = D_h(x^*, x_k) + \sum_{s=0}^{k}(f(x_s) - f(x^*))\delta$	$f(\hat{x}_k) - f(x^*) \leq O(1/\delta k)$
	$E_k = D_h(x^*, x_k) + \delta k(f(x_k) - f(x^*))$	$f(x_k) - f(x^*) \leq O(1/\delta k)$
μ-Strong Convexity $\delta > 0$	$E_k = (1 + \mu\delta)^k D_h(x^*, x_k)$	$D_h(x^*, x_k) \leq O(e^{-\mu\delta k})$
Mirror Prox:	$\frac{\nabla h(x'_{k+1}) - \nabla h(x_k)}{\delta} = -\nabla f(x_k)$, $\frac{\nabla h(x_{k+1}) - \nabla h(x_k)}{\delta} = -\nabla f(x'_{k+1})$	$\|x_k - x'_k\| = \Theta(\delta/\sigma)$
Function Class	Lyapunov Function	Convergence Rate
Convex f is (σ/δ)-smooth, h is σ-strongly convex	$E_k = D_h(x^*, x_k) + \sum_{s=0}^{k-1}(f(x'_s) - f(x^*))\delta$	$f(\hat{x}_k) - f(x^*) \leq O(1/\delta k)$
Natural Gradient Dynamic:	$\frac{d}{dt}X_t = -\nabla^2 h(X_t)^{-1}\nabla f(X_t)$	
Function Class	Lyapunov Function	Convergence Rate
Differentiable	$\mathcal{E}_t = f(X_t) - f(x^*)$	$\|\nabla f(X_t)\|^2_{X_t^*} \leq O(1/t)$
PL condition w.p μ	$\mathcal{E}_t = e^{2\mu t} f(X_t) - f(x^*)$	$f(X_t) - f(x^*) \leq O(e^{-2\mu t})$
Natural Gradient Descent:	$\frac{x_{k+1} - x_k}{\delta} = -\nabla^2 h(x_k)^{-1}\nabla f(x_k)$	
Function Class	Lyapunov Function	Convergence Rate
Differentiable f is $(1/\delta)$-smooth	$E_k = f(x_k) - f(x^*)$	$\|\nabla f(x_k)\|^2_{x_k^*} \leq O(1/\delta k)$
PL condition w.p μ f is $(1/\delta)$-smooth	$E_k = (1 - \mu\delta)^{-k}(f(x_k) - f(x^*))$	$f(x_k) - f(x^*) \leq O(e^{-\mu\delta k})$

Table 2.2: Lyapunov functions for mirror flow (MF), mirror descent (MD), the Bregman proximal minimization (BPM), mirror prox method (MPM), natural gradient flow (NGF) and natural gradient descent (NGD); with discrete-time identification $t = \delta k$, in the limit $\delta \to 0$, the results in continuous time match the results in discrete time within a factor of 2. The smoothness condition for NGD is that $D_f(x, y) \leq \frac{1}{\delta}\|x - y\|_x^2$, $\forall x, y \in \mathcal{X}$, where $\|v\|_x = \langle v, \nabla^2 h(x) v\rangle$.

Denote $\hat{X}_t = \int_0^t X_s ds/t$ as the time-average iterate. Using Jensen's inequality (A.4), we conclude $tf(\hat{X}_t) \leq \int_0^t f(X_s)ds$. By integrating (2.14) we obtain the statement, $t(f(\hat{X}_t) - f(x^*)) \leq \mathcal{E}_t \leq \mathcal{E}_0$, from which we conclude an $O(1/t)$ convergence rate,

$$f(\hat{X}_t) - f(x^*) \leq \frac{\mathcal{E}_0}{t},$$

for the optimality gap measured at the time-averaged iterate. Similar statements can be made about mirror descent and the proximal Bregman method, which are the forward and backward-Euler methods applied to (2.12), respectively.

Mirror Descent The forward-Euler method applied to MF,

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

is a stationary point of the following optimization problem,

$$x_{k+1} = \arg\min_{x \in \mathcal{X}} \left\{ \langle \nabla f(x_k), x \rangle + \frac{1}{\delta} D_h(x, x_k) \right\}. \tag{2.15}$$

As long as f is L-smooth with respect to h (A.14), where $\delta < 1/L$, then

$$E_k = D_h(x^*, x_k) + \sum_{s=0}^{k} (f(x_s) - f(x^*))\delta \tag{2.16}$$

is a Lyapunov function for mirror descent (2.15). We check,

$$\frac{E_{k+1} - E_k}{\delta} = -\left\langle \frac{\nabla h(x_{k+1}) - \nabla h(x_k)}{\delta}, x^* - x_k \right\rangle + f(x_k) - f(x^*) + \varepsilon_k^1 \overset{(2.15)}{\leq} -D_f(x^*, x_k),$$

where the error term is $\varepsilon_k^1 = f(x_{k+1}) - f(x_k) - \left\langle \frac{\nabla h(x_{k+1}) - \nabla h(x_k)}{\delta}, x_k - x_{k+1} \right\rangle - \frac{1}{\delta} D_h(x_{k+1}, x_k) = D_f(x_{k+1}, x_k) - \frac{1}{\delta} D_h(x_{k+1}, x_k)$. Take $L = 1/\delta$. We ensure the non-negativity of the error, and subsequently the upper bound, by using the $(1/\delta)$-smoothness condition with respect to h (A.14). Denote $\hat{x}_k = \delta \sum_{s=0}^k x_s/\delta k = \sum_{s=0}^k x_s/k$. Using Jensen's inequality (A.4), we conclude $\delta k f(\hat{x}_k) \leq \delta \sum_{s=0}^k f(x_s)$. By summing we obtain the statement $\delta k(f(\hat{x}_k) - f(x^*)) \leq E_k \leq E_0$, from which we conclude an $O(1/\delta k)$ convergence rate,

$$f(\hat{x}_k) - f(x^*) \leq \frac{E_0}{\delta k},$$

for the optimality gap measured at the time-averaged iterate.

Bregman proximal minimization The Bregman proximal minimization (BPM),

$$x_{k+1} = \arg\min_{x\in\mathcal{X}} \left\{ f(x) + \frac{1}{\delta} D_h(x, x_k) \right\} := \mathrm{Prox}^h_{\delta f}(x_k), \tag{2.17}$$

satisfies the variational condition

$$\frac{\nabla h(x_{k+1}) - \nabla h(x_k)}{\delta} = -\nabla f(x_{k+1}). \tag{2.18}$$

Furthermore, (2.16) is a Lyapunov function for the BPM. We check,

$$\begin{aligned} \frac{E_{k+1} - E_k}{\delta} &= -\left\langle \frac{\nabla h(x_{k+1}) - \nabla h(x_k)}{\delta}, x^* - x_{k+1} \right\rangle + f(x_{k+1}) - f(x^*) + \varepsilon_k^1 \\ &\overset{(2.18)}{\leq} -D_f(x^*, x_{k+1}), \end{aligned}$$

where the error term $\varepsilon_k^1 = -\frac{1}{\delta} D_h(x_{k+1}, x_k)$ is negative. Denote $\hat{x}_k = \delta \sum_{s=0}^k x_s / \delta k = \sum_{s=0}^k x_s / k$. By Jensen's inequality (A.4), $\delta k f(\hat{x}_k) \leq \delta \sum_{s=0}^k f(x_s)$. By summing we obtain the statement $\delta k (f(\hat{x}_k) - f(x^*)) \leq E_k \leq E_0$, from which we conclude an $O(1/\delta k)$ convergence rate,

$$f(\hat{x}_k) - f(x^*) \leq \frac{E_0}{\delta k},$$

for the optimality gap measured at the time-averaged iterate.

Mirror Prox Method The update equations for the mirror prox method (MPM) algorithm can be written,

$$x'_{k+1} \in \arg\min_{x\in\mathcal{X}} \left\{ \langle \nabla f(x_k), x \rangle + \frac{1}{\delta} D_h(x, x_k) \right\}, \tag{2.19a}$$

$$x_{k+1} \in \arg\min_{x\in\mathcal{X}} \left\{ \langle \nabla f(x'_{k+1}), x \rangle + \frac{1}{\delta} D_h(x, x_k) \right\}; \tag{2.19b}$$

the variational conditions satisfy,

$$\begin{aligned} \frac{\nabla h(x'_{k+1}) - \nabla h(x_k)}{\delta} &= -\nabla f(x_k) \\ \frac{\nabla h(x_{k+1}) - \nabla h(x_k)}{\delta} &= -\nabla f(x'_{k+1}). \end{aligned}$$

We discuss how to solve the updates (2.19) using the projection operator in Appendix B.2. We can think of this algorithm as mirror descent, where the update x_{k+1} has been replaced

with a sequence x'_{k+1}, which we use to take an additional step. To analyze MPM, we use the following Lyapunov function,

$$E_k = D_h(x^*, x_k) + \sum_{s=0}^{k} (f(x'_s) - f(x^*))\delta. \tag{2.20}$$

We check,

$$\begin{aligned}\frac{E_{k+1} - E_k}{\delta} &= -\left\langle \frac{\nabla h(x_{k+1}) - \nabla h(x_k)}{\delta}, x^* - x_{k+1} \right\rangle + f(x'_{k+1}) - f(x^*) + \varepsilon_k^1 \\ &\overset{\substack{(2.19b)\\(2.19a)}}{\leq} -D_f(x^*, x'_{k+1}) + \varepsilon_k^2 \leq -D_f(x^*, x'_{k+1}).\end{aligned}$$

Here, $\varepsilon_k^1 = -\frac{1}{\delta} D_h(x_{k+1}, x_k) = \left\langle (\nabla h(x'_{k+1}) - \nabla h(x_k))/\delta, x'_{k+1} - x_{k+1} \right\rangle - \frac{1}{\delta} D_h(x_{k+1}, x'_{k+1}) - \frac{1}{\delta} D_h(x'_{k+1}, x_k)$ and the second error, $\varepsilon_k^2 = \langle \nabla f(x'_{k+1}) - \nabla f(x_k), x'_{k+1} - x_{k+1} \rangle - \frac{1}{\delta} D_h(x_{k+1}, x'_{k+1}) - \frac{1}{\delta} D_h(x'_{k+1}, x_k)$. The last inequality, which upper bounds ε_k^2, assumes h is σ-strongly convex and f is (σ/δ)-smooth; in which case, we can use Cauchy-Schwartz (A.26), smoothness (A.13), and Young's inequality (A.25) to upper bound the inner product in ε_k^2 as follows, $\langle \nabla f(x'_{k+1}) - \nabla f(x_k), x'_{k+1} - x_{k+1} \rangle \leq \frac{\sigma}{2\delta} \|x'_{k+1} - x_k\|^2 + \frac{\sigma}{2\delta} \|x'_{k+1} - x_{k+1}\|^2$; the σ-strong convexity of h ensures this upper bound plus the remainder of the error is nonpositive. Similar to the analysis of MF, we can use Jensen's (A.4) to conclude $\delta k (f(\hat{x}'_k) - f(x^*)) \leq E_k \leq E_0$, where $\hat{x}'_k = \delta \sum_{s=0}^{k} x'_s / \delta k = \sum_{s=0}^{k} x'_s / k$, and subsequently, an $O(1/\delta k)$ convergence rate,

$$f(\hat{x}'_k) - f(x^*) \leq \frac{E_0}{\delta k},$$

on the time-averaged iterate.

The introduction of an additional iterate might seem strange, but using the σ-strong convexity of h, we can reason that $\|x_{k+1} - x'_{k+1}\| \leq (1/\sigma)\|\nabla h(x_{k+1}) - \nabla h(x'_{k+1})\| \leq (\delta/\sigma)\|\nabla f(x_k) + \nabla f(x'_{k+1})\| = \Theta(\delta/\sigma)$, where the second inequality uses (2.19); therefore in the limit $\delta \to 0$, the sequences x_{k+1} and x'_{k+1} are equivalent and we recover the mirror descent dynamic (2.12).

2.1.4 Subgradients and Time Reparameterization

We analyze the dynamic that gives rise to mirror descent algorithm, where subgradients are used instead of full gradients, in three different settings:

1. f is convex, so that $D_f^g(x, y) \geq 0 \ \forall x, y \in \mathcal{X}$ (see (A.17) for notation);
2. f is μ-strongly convex with respect h (A.7), and h is σ-strongly convex function, so that $D_f^g(y, x) \geq \mu D_h(y, x) \geq \frac{\mu\sigma}{2} \|y - x\|^2 \ \forall x, y \in \mathcal{X}$.
3. f is differentiable but has (ν, L) Hölder-continuous gradients (A.15).

MS Dynamics:	$\frac{d}{dt}\nabla h(Y_t) = -\dot{\tau}_t G_f(Y_t, \dot{Y}_t)$	$G_f(Y_t, \dot{Y}_t) \in \partial f(Y_t)$
Function Class	Lyapunov Function	Convergence Rate
Convex	$\mathcal{E}_{\tau_t} = D_h(x^*, Y_t) + \int_0^t (f(Y_s) - f(x^*))\dot{\tau}_s ds$	$f(\hat{Y}_t) - f(x^*) \leq \frac{\mathcal{E}_{\tau_0}}{\tau_t}$
μ-Strong Convexity	$\mathcal{E}_{\tau_t} = e^{\mu\tau_t} D_h(x^*, Y_t)$	$D_h(x^*, Y_t) \leq \frac{\mathcal{E}_{\tau_0}}{e^{\mu\tau_t}}$
	$\mathcal{E}_{\tau_t} = e^{\mu\tau_t} D_h(x^*, Y_t) + \frac{1}{\mu}\int_0^t (f(Y_s) - f(x^*)) de^{\mu\tau_s}$	$f(\hat{Y}_t) - f(x^*) \leq \frac{\mathcal{E}_0}{e^{\mu\tau_t}}$
MS Method:	$\frac{\nabla h(y_{k+1}) - \nabla h(y_k)}{\delta} = -\alpha_k g(y_k)$	$g(y_k) \in \partial f(y_k)$
Function Class	Lyapunov Function	Convergence Rate
Convex f is Lipschitz; h σ-strongly convex	$E_{A_k} = D_h(x^*, y_k) + \sum_{s=0}^{k-1}(f(y_s) - f(x^*))\frac{A_{s+1} - A_s}{\delta}\delta$	$f(\hat{y}_k) - f(x^*) \leq \frac{E_{A_0} + \delta\sum_{s=0}^k \varepsilon_s^1}{A_k}$
μ-Strong Convexity f is Lipschitz; h σ-strongly convex	$E_{A_k} = A_k D_h(x^*, y_k)$	$D_h(x^*, y_k) \leq \frac{E_{A_0} + \delta\sum_{s=0}^k \varepsilon_s^2}{A_k}$
	$E_{A_k} = A_k D_h(x^*, y_k) + \frac{1}{\mu}\sum_{s=0}^{k-1}(f(y_s) - f(x^*))\frac{A_{s+1} - A_s}{\delta}\delta$	$f(\hat{y}_k) - f(x^*) \leq \frac{E_{A_0} + \delta\sum_{s=0}^k \varepsilon_s^2}{A_k}$
PS Method	$\frac{\nabla h(y_{k+1}) - \nabla h(y_k)}{\delta} = -\alpha_k g(y_{k+1})$	$g(y_{k+1}) \in \partial f(y_{k+1})$
Function Class	Lyapunov Function	Convergence Rate
Convex $\delta > 0$	$E_{A_k} = D_h(x^*, y_k) + \sum_{s=0}^{k}(f(y_s) - f(x^*))\frac{A_{s+1} - A_s}{\delta}\delta$	$f(\hat{y}_k) - f(x^*) \leq \frac{E_{A_0}}{A_k}$
	$E_{A_k} = D_h(x^*, y_k) + A_k(f(y_k) - f(x^*))$	$f(y_k) - f(x^*) \leq \frac{E_{A_0}}{A_k}$
μ-Strong Convexity $\delta > 0$	$E_{A_k} = A_k D_h(x^*, y_k)$	$D_h(x^*, y_k) \leq \frac{E_{A_0}}{A_k}$
	$E_{A_k} = A_k D_h(x^*, y_k) + \frac{1}{\mu}\sum_{s=0}^{k}(f(y_s) - f(x^*))\frac{A_{s+1} - A_s}{\delta}\delta$	$f(\hat{y}_k) - f(x^*) \leq \frac{E_{A_0}}{A_k}$

Table 2.3: Lyapunov functions for the mirror descent dynamic with directional subgradients (MS Dynamic), mirror descent with subgradients (MS Method), and the proximal Bregman minimization with subgradients (PS Method). When moving to discrete time, there is a discretization error, and we choose parameters accordingly. When f is convex, $\tau_t = A_k$, so that $\dot{\tau}_t \approx (A_{k+1} - A_k)/\delta = \alpha_k$. When f is μ-strongly convex, $e^{\mu\tau_t} = A_k$, so that we have the approximation $\dot{\tau}_t = \frac{d}{dt}e^{\mu\tau_t}/\mu e^{\mu\tau_t} \approx (A_{k+1} - A_k)/\delta\mu A_{k+1} := \alpha_k$. With these choices, the errors scale as $\varepsilon_k^1 = \delta\alpha_k^2 G^2/2\sigma$ and $\varepsilon_k^2 = \delta\frac{1}{2\sigma\mu^2}\frac{\alpha_k^2}{A_{k+1}}G^2$, where $\|\partial f(x)\|_*^2 \leq G^2$. In the limit $\delta \to 0$, the discrete-time and continuous-time statements match.

When f is not smooth, the function no longer necessarily has a uniquely defined gradient at each point. One natural way around this difficulty is to use the proximal update (2.4). However, for some non-smooth functions, the update (2.4) might be too expensive to solve every iteration.

Assume f is finite, convex, and absolutely continuous on $\mathcal{X}$. Recall that the subdifferential of f at x, $\partial f(x)$, contains the subgradient of f at x, and that the directional subgradient

of f is a Borel measurable function $G_f(x, v) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$. The subdifferential and directional derivative of a function share the following relationship in this setting [64],

$$f'(x; v) = \lim_{\delta \to 0} \frac{f(x + \delta v) - f(x)}{\delta} = \sup_{g(x) \in \partial f(x)} \langle g(x), v \rangle.$$

Given f is convex, the subdifferential is nonempty, convex and compact for any x.

Similar to Su, Boyd and Candes [70], we will consider dynamical systems defined by the directional subgradient of f, when f is not smooth, with the goal of implementing this curve as an algorithm.

Before doing so, however, we discuss how to choose an appropriate scaling of time. Concretely, let $\tau : \mathbb{R} \to \mathbb{R}$ be a smooth (twice-continuously differentiable) increasing function $\dot{\tau} : \mathbb{R} \to \mathbb{R}^+$. Given a curve $X : \mathbb{R} \to \mathcal{X}$, we consider a reparameterized curve $Y = \mathbb{R} \to \mathcal{X}$ defined by,

$$Y_t = X_{\tau_t}. \tag{2.21}$$

where we adopt the shorthand, $\tau_t = \tau(t)$. That is, the new curve Y is obtained by traversing the original curve X at a new speed of time determined by τ. If $\tau_t < t$, we say that Y is the *slowed-down version* of X, because the curve Y at time t has the same value as the original curve X at the past time τ_t. We might expect that if the original curve obtained a convergence rate $f(X_t) - f(x^*) \leq O(1/t)$, the new curve Y might obtain a convergence rate $f(Y_t) - f(x^*) \leq O(1/\tau_t)$.

We study the family of curves,

$$\frac{d}{dt} \nabla h(Y_t) = -\dot{\tau}_t G_f(Y_t, \dot{Y}_t), \tag{2.22}$$

obtained by an arbitrary reparameterization of the curve, $\frac{d}{dt} \nabla h(X_t) = -G_f(X_t, \dot{X}_t)$, by the scaling (2.21), where $G_f(X_t, \dot{X}_t) \in \partial f(X_t)$. If $G_f(X_t, \dot{X}_t) = \nabla f(X_t)$ and $\tau_t = t$, then (2.22) is equivalent to the mirror descent dynamic (2.15). Let $\mathcal{X} = \mathbb{R}^n$. We provide a description of the first bullet point in the main text and provide details on the other bullet points in Appendix B.3.

2.1.4.1 Convex Functions

We analyze the family of curves (2.22) when f is convex. To do so, we apply the same time-reparameterization to the Lyapunov function (2.13),

$$\mathcal{E}_{\tau_t} = D_h(x^*, Y_t) + \int_0^t (f(Y_s) - f(x^*)) d\tau_s, \tag{2.23}$$

where $d\tau_s = \dot{\tau}_s ds$. The absolute continuity of f ensures $\mathcal{E}_{\tau_t}$ is differentiable. We check,

$$\frac{d}{dt} \mathcal{E}_{\tau_t} = -\left\langle \frac{d}{dt} \nabla h(Y_t), x^* - Y_t \right\rangle + (f(Y_t) - f(x^*)) \dot{\tau}_t \overset{(2.22)}{=} -D_f^G(x^*, Y_t) \dot{\tau}_t \leq 0.$$

Denote $\hat{Y}_t = \int_0^t Y_s d\tau_s/\tau_t$ as the time-average point. The inequality $\tau_t f(\hat{Y}_t) \leq \int f(Y_s) d\tau_s$ follows from applying Jensen's inequality (A.4). By integrating the Lyapunov function, we obtain the statement $\tau_t(f(\hat{Y}_t) - f(x^*)) \leq \mathcal{E}_{\tau_t} \leq \mathcal{E}_{\tau_0}$ for the curve. Therefore we can conclude a $O(1/\tau_t)$ convergence rate for (2.22) on the time-averaged iterate:

$$f(\hat{Y}_t) - f(x^*) \leq \frac{\mathcal{E}_{\tau_0}}{\tau_t}.$$

We apply a similar argument to the discretizations of (2.22) using a discretization of the Lyapunov function (2.23). Make the identifications $A_k := \tau_t$ and $\alpha_k = \frac{A_{k-1} - A_k}{\delta} := \dot{\tau}_t$. Different scalings produce errors with differing scales in discrete time. At the end of the discrete-time analysis, the scaling which maximizes the upper-bound is chosen.

Mirror subgradient method The forward-Euler discretization of (2.22),

$$y_{k+1} = \arg\min_{x \in \mathbb{R}^d} \left\{ \alpha_k \langle g(y_k), x \rangle + \frac{1}{\delta} D_h(x, y_k) \right\}, \tag{2.24}$$

chooses an element of the subdifferential at every iteration $g(y_k) \in \partial f(y_k)$. It satisfies the variational condition $\frac{\nabla h(y_{k+1}) - \nabla h(y_k)}{\delta} = -\alpha_k g(y_k)$. We analyze (2.24) using the Lyapunov function,

$$E_{A_k} = D_h(x^*, y_k) + \sum_{s=0}^{k-1} (f(y_s) - f(x^*)) \frac{A_{s+1} - A_s}{\delta} \delta. \tag{2.25}$$

We check,

$$\begin{aligned} \frac{E_{A_{k+1}} - E_{A_k}}{\delta} &= -\left\langle \frac{\nabla h(y_{k+1}) - \nabla h(y_k)}{\delta}, x^* - y_k \right\rangle + (f(y_k) - f(x^*))\alpha_k + \varepsilon_k^1 \\ &\overset{(2.24)}{=} -D_f^g(x^*, y_k)\alpha_k + \varepsilon_k^1 \end{aligned}$$

where the error scales as $\varepsilon_k^1 = \alpha_k \langle g(y_k), y_k - y_{k+1} \rangle - \frac{1}{\delta} D_h(y_{k+1}, y_k)$. Define the time-averaged iterate $\hat{y}_k = \delta \sum_{s=0}^k y_s \alpha_s / A_k$. By summing, we obtain the statement $A_k(f(\hat{y}_k) - f(x^*)) \leq E_{A_k} \leq E_{A_0} + \delta \sum_{s=0}^k \varepsilon_s^1$, as well as the bound,

$$f(\hat{y}_k) - f(x^*) \leq \frac{E_{A_0} + \delta \sum_{s=0}^k \varepsilon_s^1}{A_k}. \tag{2.26}$$

The bound provides us with a rate of convergence as long as $\sum_{s=0}^k \varepsilon_s^1 / A_k \to 0$ as $k \to \infty$. To obtain an upper bound on the error, we typically assume that h is σ-strongly convex (A.6); with this assumption, Young's inequality (A.25) can be used to obtain the following upper bound $\varepsilon_k^1 \leq \delta \frac{\alpha_k^2}{2\sigma} \|g(y_k)\|_*^2$. Assume f is Lipschitz on $\mathcal{X}$, so that for all $y \in \mathcal{X}$, $\|\partial f(y)\|_*^2 \leq G^2$, for some finite constant G^2 (see (A.17) for notation). Maximizing the bound, $\delta \sum_{s=0}^k \frac{\alpha_s^2}{2\sigma} G^2 / A_k$ over the sequence A_k leads to the choice $\alpha_K = D_h(x^*, X_0)/G^2\sqrt{K}$, and convergence rate $O(1/\sqrt{K})$. There is a matching lower bound for the subgradient oracle function.

Proximal Bregman method The backward-Euler discretization of (2.22)

$$y_{k+1} = \arg\min_{x\in\mathbb{R}^d} \left\{ \alpha_k f(x) + \frac{1}{\delta} D_h(x, y_k) \right\}, \tag{2.27}$$

satisfies variational inequality we can write as $\frac{\nabla h(y_{k+1}) - \nabla h(y_k)}{\delta} = -\alpha_k g(y_{k+1})$, where $g(y_{k+1}) \in \partial f(y_{k+1})$. We can similarly be analyzed using Lyapunov function (2.25), as well as the Lyapunov function

$$E_{A_k} = D_h(x^*, y_k) + A_k(f(y_k) - f(x^*)). \tag{2.28}$$

We check,

$$\begin{aligned}\frac{E_{A_{k+1}} - E_{A_k}}{\delta} &= -\left\langle \frac{\nabla h(y_{k+1}) - \nabla h(y_k)}{\delta}, x^* - y_{k+1} \right\rangle + \alpha_k f(y_{k+1}) - f(x^*) \\ &\quad + A_k \frac{f(y_{k+1}) - f(y_k)}{\delta} + \varepsilon_k^1 \leq -D_f^g(x^*, y_{k+1})\alpha_k + \varepsilon_k^2\end{aligned}$$

where the errors $\varepsilon_k^1 = -\frac{1}{\delta} D_h(y_{k+1}, y_k)$ and $\varepsilon_k^2 = \varepsilon_k^1 + A_k \frac{f(y_{k+1}) - f(y_k)}{\delta}$ are negative. The non-negativity of the second error follows from the descent property of the proximal method (2.27), i.e. $A_k \frac{f(y_{k+1}) - f(y_k)}{\delta} \leq -\frac{A_k}{\alpha_k} \frac{1}{\delta^2} D_h(y_{k+1}, y_k)$.

2.1.5 Dual Averaging Dynamic

We analyze the dynamic that gives rise to the dual averaging algorithm in the setting when f is convex.

Let $\mathcal{X} = \mathbb{R}^d$ and $f : \mathbb{R}^d \to \mathbb{R}$ be an absolutely continuous convex function. Take $\dot\gamma_t, \gamma_t, \dot\tau_t, \tau_t > 0$. The dual averaging dynamic is given by the system of equations,

$$\frac{d}{dt} Y_t = -\dot\tau_t G(X_t, \dot X_t) \tag{2.29a}$$

$$Y_t = \gamma_t \nabla h(X_t). \tag{2.29b}$$

Here, $G_f(X_t, \dot X_t) \in \partial f(X_t)$ is a directional subgradient of f at X_t and we choose $h : \mathcal{X} \to \mathbb{R}$ to be a σ-storngly convex function with a well-defined prox-center [44]; specifically, the prox-center y is defined as the solution to the optimization problem,

$$y = \arg\min_{x\in\mathcal{X}} h(x). \tag{2.30}$$

It is usually taken without loss of generality that $h(y) = 0$ so that $h(x) = D_h(x, y) \geq 0$, $\forall x \in \mathcal{X}$. In particular, when $h(x) = \frac{1}{2}\|x\|^2$, then the point $y = 0$ is the prox-center. If we choose our initial position X_0 to be the prox-center (2.30), the function,

$$\mathcal{E}_t = \gamma_t D_h(x^*, X_t) + \int_0^t (f(X_s) - f(x^*)) d\tau_s,$$

DA Dynamic:	$\frac{d}{dt}Y_t = -\dot{\tau}_t G_f(X_t, \dot{X}_t),\ Y_t = \gamma_t \nabla h(X_t)$	$G_f(X_t, \dot{X}_t) \in \partial f(X_t)$
Function Class	Lyapunov Function	Convergence Rate
Convex	$\mathcal{E}_{\tau_t} = \gamma_t D_h(x^*, X_t) + \int_0^t (f(X_s) - f(x^*))\dot{\tau}_s ds$	$f(\hat{X}_t) - f(x^*) \leq \frac{\gamma_t D_h(x^*, X_0)}{\tau_t}$
DA Algorithm:	$\frac{y_{k+1}-y_k}{\delta} = -\alpha_k g(x_k),\ y_k = \gamma_k \nabla h(x_k)$	$g(x_k) \in \partial f(x_k)$
Function Class	Lyapunov Function	Convergence Rate
Convex f is Lipschitz; h σ-strongly convex	$E_k = \gamma_k D_h(x^*, x_k) + \sum_{s=0}^{k-1}(f(x_s) - f(x^*))\alpha_s \delta$	$f(\hat{x}_k) - f(x^*) \leq \frac{\gamma_k D_h(x^*, x_0) + \delta \sum_{s=0}^k \varepsilon_s^1}{A_k}$
Proximal DA:	$\frac{y_{k+1}-y_k}{\delta} = -\alpha_k g(x_{k+1}),\ y_k = \gamma_k \nabla h(x_k)$	$g(x_{k+1}) \in \partial f(x_{k+1})$
Function Class	Lyapunov Function	Convergence Rate
Convex $\delta > 0$	$E_k = \gamma_k D_h(x^*, x_k) + \sum_{s=0}^{k}(f(x_s) - f(x^*))\alpha_s \delta$	$f(\hat{x}_k) - f(x^*) \leq \frac{\gamma_k D_h(x^*, x_0)}{A_k}$

Table 2.4: Lyapunov functions for the dual averaging (DA) dynamic, dual averaging (DA) algorithm , and the backward-Euler approximation of the dual averaging dynamics (proximal DA); for the dual averaging algorithm, $\alpha_k = \frac{A_{k+1}-A_k}{\delta}$, $\varepsilon_k^1 = \delta \frac{1}{2\sigma}\frac{\alpha_k^2}{\gamma_k} G^2$ where $\|\partial f(x)\|_*^2 \leq G^2$. In the limit $\delta \to 0$, the discrete-time and continuous-time statements match.

provides a rate of convergence for (2.29). We check,

$$\begin{aligned}\frac{d}{dt}\mathcal{E}_t &= D_h(x^*, X_t)\frac{d}{dt}\gamma_t - \gamma_t \left\langle \frac{d}{dt}\nabla h(X_t), x^* - X_t \right\rangle + \dot{\tau}_t(f(X_t) - f(x^*)) \\ &\overset{(2.29b)}{=} (h(x^*) - h(X_t))\frac{d}{dt}\gamma_t - \left\langle \frac{d}{dt}Y_t, x^* - X_t \right\rangle + \dot{\tau}_t(f(X_t) - f(x^*)) \\ &\overset{(2.29a)}{=} -\dot{\tau}_t D_f^G(x^*, X_t) + \dot{\gamma}_t(h(x^*) - h(X_t)) \leq \dot{\gamma}_t D_h(x^*, X_0).\end{aligned}$$

The last inequality uses the fact that $h(x) = D_h(x, X_0) \geq 0, \forall x \in \mathcal{X}$ as well as the definition of a prox-center $h(x^*) = D_h(x^*, X_0)$. Denote $\hat{X}_t = \int_0^t X_s d\tau_s / \tau_t$ as the time-average iterate and note that the inequality $\tau_t f(\hat{X}_t) \leq \int_0^t f(X_s) d\tau_s$ follows from Jensen's (A.4). By integrating the Lyapunov argument $\frac{d}{dt}\mathcal{E}_t \leq \dot{\gamma}_t D_h(x^*, X_0)$, we obtain the statement, $\tau_t(f(\hat{X}_t) - f(x^*)) \leq \mathcal{E}_t \leq \mathcal{E}_0 + \gamma_t D_h(x^*, X_0) - \gamma_0 D_h(x^*, X_0)$, from which we conclude an $O(\gamma_t/\tau_t)$ convergence rate,

$$f(\hat{X}_t) - f(x^*) \leq \frac{\mathcal{E}_0 + \gamma_t D_h(x^*, X_0)}{\tau_t}.$$

Dual Averaging The dual averaging algorithm,

$$x_{k+1} = \arg\min_{x \in \mathcal{X}} \left\{ \delta \sum_{s=0}^{k} \alpha_s \langle g(x_s), x \rangle + \gamma_k D_h(x, x_0) \right\}, \tag{2.31}$$

satisfies the variational condition $\frac{\gamma_{k+1}\nabla h(x_{k+1})-\gamma_k\nabla h(x_k)}{\delta} = -\alpha_k g(x_k)$, where x_0 is the prox-center (2.30), $\alpha_k = \frac{A_{k+1}-A_k}{\delta}$ and $g(x_k) \in \partial f(x_k)$. Denote $y_k = \gamma_k \nabla h(x_k)$. The variational condition can be written, $\frac{y_{k+1}-y_k}{\delta} = -\alpha_k g(x_k)$. Using the discrete-time function,

$$E_k = \gamma_k D_h(x^*, x_k) + \sum_{s=0}^{k-1}(f(x_s) - f(x^*))\frac{A_{s+1}-A_s}{\delta}\delta,$$

we check,

$$\begin{aligned}
\frac{E_{k+1}-E_k}{\delta} &= D_h(x^*, x_{k+1})\frac{\gamma_{k+1}-\gamma_k}{\delta} - \gamma_k\left\langle \frac{\nabla h(x_{k+1})-\nabla h(x_k)}{\delta}, x^* - x_k\right\rangle \\
&\quad + \alpha_k(f(x_k) - f(x^*)) + \varepsilon_k^1 \\
&= (h(x^*) - h(x_{k+1}))\frac{\gamma_{k+1}-\gamma_k}{\delta} - \left\langle \frac{y_{k+1}-y_k}{\delta}, x^* - x_k\right\rangle + \alpha_k(f(x_k) - f(x^*)) + \varepsilon_k^1 \\
&\overset{(2.31)}{=} -\alpha_k D_f^g(x^*, x_k) + \frac{\gamma_{k+1}-\gamma_k}{\delta}(h(x^*) - h(x_{k+1})) + \varepsilon_k^2 \leq \frac{\gamma_{k+1}-\gamma_k}{\delta} D_h(x_*, x_0) + \varepsilon_k^1.
\end{aligned}$$

Here, the error scales as $\varepsilon_k^1 = \alpha_k\langle g(x_k), x_k - x_{k+1}\rangle - \frac{\gamma_k}{\delta}D_h(x_{k+1}, x_k)$. The final upper bound follows from noting $-D_f^g(x^*, x_k) \leq 0$ and using the definition of the prox-center. Assume $\|\partial f(y_k)\|_*^2 \leq G^2$ for all $y_k \in \mathcal{X}$ and some constant G. Using the σ-strong convexity of h, we can use Young's inequality to upper bound the error $\varepsilon_k^2 \leq \frac{\alpha_k^2\delta}{2\sigma\gamma_k}G := \varepsilon_k^3$. Denote $\hat{x}_k = \delta\sum_{s=0}^k x_s\alpha_s/A_k$ as the time-average iterate and note that the inequality $A_k f(\hat{x}_k) \leq \delta\sum_{s=0}^k f(x_s)\alpha_s$ follows from Jensen's (A.4). By summing the Lyapunov function we obtain the statement, $A_k(f(\hat{x}_k) - f(x^*)) \leq E_k \leq E_0 + \gamma_k D_h(x^*, x_0) - \gamma_0 D_h(x^*, x_0) + \delta\sum_{s=0}^k \varepsilon_i^3$, from which we obtain the convergence bound,

$$f(\hat{x}_k) - f(x^*) \leq \frac{E_0 + \gamma_k D_h(x^*, x_0) + \delta^2\frac{1}{2\sigma}\sum_{s=0}^k \frac{\alpha_s^2}{\gamma_s}G^2}{A_k}.$$

If we assume with out loss of generality $\sigma = 1$, and choose $A_k = k$, $\delta = 1$ and $\gamma_k = \frac{G^2}{D_h(x^*,x_0)}\sqrt{k+1}$, we obtain $O(1/\sqrt{k})$ convergence rate [44]. This bound matches the oracle function lower bound for algorithms designed using only subgradients of convex functions (i.e. is provably optimal). Furthermore, as $\delta \to 0$, the error $\varepsilon_k^3 \to 0$ and we recover the result for the continuous time dynamics.

2.1.6 Conditional Gradient Dynamic

We analyze the dynamical systems that gives rise to the conditional gradient method (Frank-Wolfe algorithm) in two different settings:

1. $\mathcal{X}$ is a convex, compact set, f is Lipschitz on $\mathcal{X}$ and has $(1/\epsilon)$-smooth gradients (A.13).

2. $\mathcal{X}$ is a convex, compact set, f is Lipschitz on $\mathcal{X}$ and has $(\nu, 1/\epsilon)$ Hölder-continuous gradients (A.15).

Conditional Gradient Dynamic:	$\langle \nabla f(X_t), x - Z_t\rangle, \quad \forall x \in \mathcal{X},$	$\dot{X}_t = \frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}}(Z_t - X_t)$
Function Class	Lyapunov Function	Convergence Rate
Convex $\mathcal{X}$ is compact and convex, f is Lipschitz on $\mathcal{X}$	$\mathcal{E}_t = e^{\beta_t}(f(X_t) - f(x^*))$	$f(X_t) - f(x^*) \leq \frac{\mathcal{E}_0}{e^{\beta_t}}$
Conditional Gradient Algorithm	$\langle \nabla f(x_k), x - z_k\rangle, \quad \forall x \in \mathcal{X},$	$\frac{x_{k+1}-x_k}{\delta} = \tau_k(z_k - x_k)$
Function Class	Lyapunov Function	Convergence Rate
Convex $\mathcal{X}$ is compact and convex, f is $(1/\delta)$-smooth	$E_k = A_k(f(x_k) - f(x^*))$	$f(x_k) - f(x^*) \leq \frac{E_0 + \delta\sum_{s=0}^{k}\varepsilon_s^1}{A_k}$

Table 2.5: Lyapunov functions for conditional gradient descent (CGD) dynamic and the conditional gradient descent (CGD) algorithm. Here, $\frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}} \approx \frac{A_{k+1}-A_k}{\delta A_{k+1}} := \tau_k$, $\varepsilon_{k+1} = \delta\frac{A_{k+1}\tau_k^2}{2\epsilon}\left\|z_k - x_k\right\|^2$. In the limit $\delta \to 0$, discrete-time and continuous-time statements match.

The conditional gradient dynamic,

$$\frac{d}{dt}X_t = \frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}}(Z_t - X_t), \tag{2.32a}$$

$$Z_t \in \arg\min_{v\in\mathcal{X}}\langle \nabla f(X_t), v\rangle \tag{2.32b}$$

is defined on convex, compact sets $\mathcal{X}$. The update (2.32b) satisfies the variational condition $0 \leq \langle \nabla f(X_t), x - Z_t\rangle, \forall x \in \mathcal{X}$. This dynamical system is remarkably similar to the dynamical system (2.38), where instead of using the Bregman divergence to ensure nonnegativity of the variational inequality $0 \leq \langle \nabla f(X_t), x - Z_t\rangle \frac{d}{dt}e^{\beta_t}$, we simply assume (2.32b) holds on the domain $\mathcal{X}$. The following function,

$$\mathcal{E}_t = e^{\beta_t}(f(X_t) - f(x)), \tag{2.33}$$

is a Lyapunov function for (2.32). We check,

$$\begin{aligned}\frac{d}{dt}\mathcal{E}_t &= e^{\beta_t}\frac{d}{dt}f(X_t) + (f(X_t) - f(x^*))\frac{d}{dt}e^{\beta_t}\\ &\leq e^{\beta_t}\langle \nabla f(X_t), \dot{X}_t\rangle + \langle \nabla f(X_t), x^* - X_t\rangle\frac{d}{dt}e^{\beta_t}\\ &\overset{(2.32a)}{=} \langle \nabla f(X_t), x^* - Z_t\rangle\frac{d}{dt}e^{\beta_t} \overset{(2.32b)}{\leq} 0\end{aligned}$$

Applying the backward-Euler scheme to (2.32a) and (2.32b), with the same approximations, $\frac{d}{dt}X_t = \frac{x_{k+1}-x_k}{\delta}$, $\frac{d}{dt}e^{\beta t} = \frac{A_{k+1}-A_k}{\delta}$, and denoting $\tau_k = \frac{A_{k+1}-A_k}{\delta A_{k+1}}$, we obtain the variational conditions for the following algorithm:

$$z_k = \arg\min_{z\in\mathcal{X}} \langle \nabla f(x_k), z\rangle, \tag{2.34a}$$

$$\frac{x_{k+1}-x_k}{\delta} = \tau_k(z_k - x_k). \tag{2.34b}$$

We can write update (2.34b) as $x_{k+1} = \delta\tau_k z_k + (1-\delta\tau_k)x_k$Update (2.34a) requires the assumptions that $\mathcal{X}$ be convex and compact; under this assumption, (2.34a) satisfies

$$0 \le \langle \nabla f(x_k), x - z_k\rangle, \forall x \in \mathcal{X},$$

consistent with (2.32b). The following proposition describes how a discretization of (2.33) can be used to analyze the behavior of algorithm (2.34). Assume f is convex and $\mathcal{X}$ is convex and compact. If f is $(1/\epsilon)$-smooth, using the Lyapunov function,

$$E_k = A_k(f(x_k) - f(x^*)), \tag{2.35}$$

we obtain the error bound,

$$\begin{aligned}
\frac{E_{k+1}-E_k}{\delta} &= A_{k+1}\frac{f(x_{k+1})-f(x_k)}{\delta} + (f(x_k)-f(x^*))\frac{A_{k+1}-A_k}{\delta} \\
&\le A_{k+1}\left\langle \nabla f(x_k), \frac{x_{k+1}-x_k}{\delta}\right\rangle + \langle \nabla f(x_k), x^*-x_k\rangle\frac{A_{k+1}-A_k}{\delta} + \varepsilon_{k+1} \\
&\overset{(2.34b)}{=} \langle \nabla f(x_k), x^*-z_k\rangle\frac{A_{k+1}-A_k}{\delta} + \varepsilon_{k+1} \overset{(2.34a)}{\le} \varepsilon_k^1.
\end{aligned}$$

The second inequality uses the convexity and $(1/\epsilon)$-smoothness of f. The error scales as $\varepsilon_{k+1} = \delta\frac{A_{k+1}}{2\epsilon}\|\frac{x_{k+1}-x_k}{\delta}\|^2 = \delta\frac{A_{k+1}\tau_k^2}{2\epsilon}\|z_k - x_k\|^2$. If instead we assume f has (ϵ,ν)-Hölder-continuous gradients (A.15), the error in algorithm (2.34) now scales as $\varepsilon_{k+1} = \delta\frac{A_{k+1}}{2\epsilon}\|\frac{x_{k+1}-x_k}{\delta}\|^{1+\nu}$ $= \delta^\nu\frac{A_{k+1}\tau_k^{1+\nu}}{(1+\nu)\epsilon}\|z_k - x_k\|^{1+\nu}$. Taking $A_k = \frac{(k+1)(k+2)}{2}$ we infer the convergence rates $O(1/\epsilon k)$ and $O(1/\epsilon k^\nu)$, respectively.

2.2 Lyapunov Analysis of Second-Order Dynamics

This section is based on the work *A Lyapunov analysis of momentum methods in optimization.* A. Wilson, B. Recht and M. I. Jordan. *Submitted to Mathematics of Operations Research (MOOR), 2016.*

2.2.1 A Lyapunov Analysis of Momentum Methods in Optimization

Momentum is a powerful heuristic for accelerating the convergence of optimization methods. One can intuitively "add momentum" to a method by adding to the current step a weighted version of the previous step, encouraging the method to move along search directions that had been previously seen to be fruitful. Such methods were first studied formally by Polyak [54], and have been employed in many practical optimization solvers. As an example, since the 1980s, momentum methods have been popular in neural networks as a way to accelerate the backpropagation algorithm. The conventional intuition is that momentum allows local search to avoid "long ravines" and "sharp curvatures" in the sublevel sets of cost functions [66].

Polyak motivated momentum methods by an analogy to a "heavy ball" moving in a potential well defined by the cost function. However, Polyak's physical intuition was difficult to make rigorous mathematically. For quadratic costs, Polyak was able to provide an eigenvalue argument that showed that his Heavy Ball Method required no more iterations than the method of conjugate gradients [54].[1] Despite its intuitive elegance, however, Polyak's eigenvalue analysis does not apply globally for general convex cost functions. In fact, Lessard *et al.* derived a simple one-dimensional counterexample where the standard Heavy Ball Method does not converge [30].

In order to make momentum methods rigorous, a different approach was required. In celebrated work, Nesterov devised a general scheme to accelerate convex optimization methods, achieving optimal running times under oracle models in convex programming [43]. To achieve such general applicability, Nesterov's proof techniques abandoned the physical intuition of Polyak [43]; in lieu of differential equations and Lyapunov functions, Nesterov devised the method of *estimate sequences* to verify the correctness of these momentum-based methods. Researchers have struggled to understand the foundations and scope of the estimate sequence methodology since Nesterov's initial papers. The associated proof techniques are often viewed as an "algebraic trick."

To overcome the lack of fundamental understanding of the estimate sequence technique, several authors have recently proposed schemes to achieve acceleration without appealing to it [15, 9, 30, 14]. One promising general approach to the analysis of acceleration has been to analyze the continuous-time limit of accelerated methods [70, 76, 25], or to derive these limiting ODEs directly via an underlying Lagrangian [76], and to prove that the ODEs are stable via a Lyapunov function argument. However, these methods stop short of providing principles for deriving a discrete-time optimization algorithm from a continuous-time ODE. There are many ways to discretize ODEs, but not all of them give rise to convergent methods or to acceleration. Indeed, for unconstrained optimization on Euclidean spaces in the setting where the objective is strongly convex, Polyak's Heavy Ball method and Nesterov's accelerated gradient descent have the same continuous-time limit. One recent line of attack on the discretization problem is via the use of a time-varying Hamiltonian and symplectic

[1] Indeed, when applied to positive-definite quadratic cost functions, Polyak's Heavy Ball Method is equivalent to Chebyshev's Iterative Method [10].

integrators [52]. In this chapter, we present a different approach, one based on a fuller development of Lyapunov theory. In particular, we present Lyapunov functions for both the continuous and discrete settings, and we show how to move between these Lyapunov functions. Our Lyapunov functions are time-varying and they thus allow us to establish rates of convergence. They allow us to dispense with estimate sequences altogether, in favor of a dynamical-systems perspective that encompasses both continuous time and discrete time.

2.2.1.1 The Bregman Lagrangian

We [4] introduced the following function on curves,

$$\mathcal{L}(x, v, t) = e^{\alpha_t + \gamma_t} \left(D_h \left(x, x + e^{-\alpha_t} v \right) - e^{\beta_t} f(x) \right), \tag{2.36}$$

where $x \in \mathcal{X}$, $v \in \mathbb{R}^d$, and $t \in \mathbb{R}$ represent position, velocity and time, respectively [76]. They called (2.36) the *Bregman Lagrangian*. The functions $\alpha, \beta, \gamma : \mathbb{R} \to \mathbb{R}$ are arbitrary smooth increasing functions of time that determine the overall damping of the Lagrangian functional, as well as the weighting on the velocity and potential function. We also introduced the following "ideal scaling conditions," which are needed to obtain optimal rates of convergence:

$$\dot{\gamma}_t = e^{\alpha_t} \tag{2.37a}$$

$$\dot{\beta}_t \leq e^{\alpha_t}. \tag{2.37b}$$

Given $\mathcal{L}(x, v, t)$, we can define a functional on curves $\{X_t : t \in \mathbb{R}\}$ called the *action* via integration of the Lagrangian: $\mathcal{A}(X) = \int_{\mathbb{R}} \mathcal{L}(X_t, \dot{X}_t, t) dt$. Calculation of the Euler-Lagrange equation, $\frac{\partial \mathcal{L}}{\partial x}(X_t, \dot{X}_t, t) = \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial v}(X_t, \dot{X}_t, t)$, allows us to obtain a stationary point for the problem of finding the curve which minimizes the action. We showed [76, (2.7)] that under the first scaling condition (2.37a), the Euler-Lagrange equation for the Bregman Lagrangian reduces to the following ODE:

$$\frac{d}{dt} \nabla h(X_t + e^{-\alpha_t} \dot{X}_t) = -e^{\alpha_t + \beta_t} \nabla f(X_t). \tag{2.38}$$

Second Bregman Lagrangian. We [4] introduced a second function on curves,

$$\mathcal{L}(x, v, t) = e^{\alpha_t + \gamma_t + \beta_t} \left(\mu D_h \left(x, x + e^{-\alpha_t} v \right) - f(x) \right), \tag{2.39}$$

using the same definitions and scaling conditions. The Lagrangian (2.39) places a different damping on the kinetic energy than in the original Bregman Lagrangian (2.36).

Proposition 2.2.1. *Under the same scaling condition* (2.37a)*, the Euler-Lagrange equation for the second Bregman Lagrangian* (2.39) *reduces to:*

$$\frac{d}{dt} \nabla h(X_t + e^{-\alpha_t} \dot{X}_t) = \dot{\beta}_t \nabla h(X_t) - \dot{\beta}_t \nabla h(X_t + e^{-\alpha_t} \dot{X}_t) - \frac{e^{\alpha_t}}{\mu} \nabla f(X_t). \tag{2.40}$$

We provide a proof of Proposition 2.2.1 in Appendix B.5.1.

Summary We summarize the results presented in this subsection in Table 2.6.

Accelerated Dynamic 1	$\frac{d}{dt}\nabla h(Z_t) = -\left(\frac{d}{dt}e^{\beta_t}\right)\nabla f(X_t)$	$\frac{d}{dt}X_t = \frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}}(Z_t - X_t)$
Function Class	Lyapunov Function	Convergence Rate
Convex	$\mathcal{E}_t = D_h(x^*, Z_t) + e^{\beta_t}(f(X_t) - f(x^*))$	$f(X_t) - f(x^*) \leq O(1/e^{\beta_t})$
Accelerated Algorithm 1	$\frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta} = -\alpha_k \nabla f(x_{k+1})$	$\frac{x_{k+1} - y_k}{\delta} = \tau_k(z_k - y_k)$
	$\|y_k - x_k\| = O(\delta)$, $\delta = \sqrt{\epsilon\sigma}$	$y_k = x_k - \delta\nabla f(x_k)$
Function Class	Lyapunov Function	Convergence Rate
Convex f is $(1/\epsilon)$-smooth, h is σ-strongly convex	$E_k = D_h(x^*, z_k) + A_k(f(y_k) - f(x^*))$	$f(y_k) - f(x^*) \leq O(1/A_k)$
Accelerated Mirror Prox	$\frac{\nabla h(z'_{k+1}) - \nabla h(z_k)}{\delta} = -\alpha_k \nabla f(x'_{k+1})$, $\delta = \sqrt{\epsilon\sigma}$	$\frac{x_{k+1} - x_k}{\delta} = \tau_k(z'_{k+1} - x_k)$
	$\frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta} = -\alpha_k \nabla f(x_{k+1})$, $\|x'_k - x_k\| = O(\delta)$,	$\frac{x_{k+1} - x_k}{\delta} = \tau_k(z_k - x_k)$
Function Class	Lyapunov Function	Convergence Rate
Convex f is $(1/\epsilon)$-smooth, h is σ-strongly convex	$E_k = D_h(x^*, z_k) + A_k(f(y_k) - f(x^*))$	$f(y_k) - f(x^*) \leq O(1/A_k)$
Accelerated Dynamic 2	$\frac{d}{dt}\nabla h(Z_t) = \frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}}\left(\nabla h(X_t) - \nabla h(Z_t) - \frac{1}{\mu}\nabla f(X_t)\right)$	$\dot{X}_t = \frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}}(Z_t - X_t)$
Function Class	Lyapunov Function	Convergence Rate
f is μ-*uniformly convex w.r.t* h	$\mathcal{E}_t = e^{\beta_t}\left(\mu D_h(x^*, Z_t) + f(X_t) - f(x^*)\right)$	$f(X_t) - f(x^*) \leq O(1/e^{\beta_t})$
Accelerated Algorithm 2	$\frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta} = \tau_k\left(\nabla h(x_k) - \nabla h(z_k) - \frac{1}{\mu}\nabla f(x_k)\right)$	$\frac{x_{k+1} - y_{k+1}}{\delta} = \tau_k(z_{k+1} - x_{k+1})$
	$\|y_{k+1} - x_k\| = O(\delta)$, $\delta = \sqrt{\epsilon}$	$y_{k+1} = x_k - \delta\nabla f(x_k)$
Function Class	Lyapunov Function	Convergence Rate
μ-Strongly Convex f is $(1/\epsilon)$-smooth, h is Euclidean	$E_k = A_k(D_h(x^*, z_k) + f(y_k) - f(x^*))$	$f(y_k) - f(x^*) \leq O(1/A_k)$

Table 2.6: Lyapunov functions for accelerated mirror descent (AMD) dynamic, accelerated mirror descent (AMD), accelerated mirror prox (AMP), and the backward Euler discretization. For AMD1 and AMP, we take $A_{k+1} = \frac{\sigma\epsilon(k+1)(k+2)}{4}$, $\alpha_k = \frac{A_{k+1} - A_k}{\delta} = \frac{\sqrt{\sigma\epsilon}(k+2)}{2}$, $\delta = \sqrt{\epsilon\sigma}$ and for AMD2, we take $A_{k+1} = (1 - \sqrt{\mu}\delta)^{-(k+1)}$, $\tau_k = \frac{A_{k+1} - A_k}{\delta A_{k+1}} = \sqrt{\mu}$, $\delta = \sqrt{\epsilon}$.

2.2.1.2 Methods arising from the first Euler-Lagrange equation

Assume f is convex, h is strictly convex, and the second ideal scaling condition (2.37b) holds with equality. We write the Euler Lagrange equation (2.38) as,

$$\frac{d}{dt}X_t = \frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}}(Z_t - X_t), \tag{2.41a}$$

$$\frac{d}{dt}\nabla h(Z_t) = -\nabla f(X_t)\frac{d}{dt}e^{\beta_t}. \tag{2.41b}$$

For this continuous-time dynamical system,

$$\mathcal{E}_t = D_h(x^*, Z_t) + e^{\beta_t}(f(X_t) - f(x^*)) \tag{2.42}$$

is a Lyapunov function. We check,

$$\begin{aligned}\frac{d}{dt}\mathcal{E}_t &= -\left\langle \frac{d}{dt}\nabla h(Z_t), x^* - Z_t \right\rangle + e^{\beta_t}\frac{d}{dt}f(X_t) + (f(X_t) - f(x^*))\frac{d}{dt}e^{\beta_t}. \\ &\overset{(2.41b)}{=} \left(\langle \nabla f(X_t), x^* - Z_t\rangle + f(X_t) - f(x^*)\right)\frac{d}{dt}e^{\beta_t} + e^{\beta_t}\frac{d}{dt}f(X_t) \\ &\overset{(2.41a)}{=} -D_f(x^*, X_t)\frac{d}{dt}e^{\beta_t} \leq 0\end{aligned} \tag{2.43}$$

This argument allows us to conclude a $O(e^{-\beta_t})$ convergence rate.

Backward-Backward-Euler. Written as an algorithm, the backward Euler method applied to (2.41a) and (2.41b) has the following update equations:

$$z_{k+1} = \underset{\substack{z\in\mathcal{X} \\ x=\frac{\delta\tau_k}{1+\delta\tau_k}z+\frac{1}{1+\delta\tau_k}x_k}}{\arg\min} \left\{ A_k f(x) + \frac{1}{\delta\tau_k}D_h\left(z, z_k\right)\right\}, \tag{2.44a}$$

$$x_{k+1} = \frac{\delta\tau_k}{1+\delta\tau_k}z_{k+1} + \frac{1}{1+\delta\tau_k}x_k; \tag{2.44b}$$

these updates satisfy the variational conditions $\frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta} = -\nabla f(x_{k+1})\frac{A_{k+1}-A_k}{\delta}$ and $\frac{x_{k+1}-x_k}{\delta} = \tau_k(z_{k+1} - x_{k+1})$ where $\tau_k = \frac{A_{k+1}-A_k}{\delta A_k}$, respectively. Let $\alpha_k = \frac{A_{k+1}-A_k}{\delta}$. We now state our main proposition for the discrete-time dynamics.

Proposition 2.2.2. *Using the discrete-time Lyapunov function,*

$$E_k = D_h(x^*, z_k) + A_k(f(x_k) - f(x^*)), \tag{2.45}$$

the bound $\frac{E_{k+1}-E_k}{\delta} \leq 0$ *holds for algorithm* (2.44).

This allows us to conclude a general $O(1/A_k)$ convergence rate for the implicit method (2.44). Using these identities, we have the following derivation:

$$\begin{aligned}\frac{E_{k+1}-E_k}{\delta} &= -\left\langle \frac{\nabla h(z_{k+1})-\nabla h(z_k)}{\delta}, x^*-z_{k+1}\right\rangle + A_k\frac{f(x_{k+1})-f(x_k)}{\delta}\\ &\quad + (f(x_{k+1})-f(x^*))\alpha_k + \varepsilon_k^1\\ &\overset{(2.44a)}{=} (\langle \nabla f(x_{k+1}), x^*-z_{k+1}\rangle + f(x_{k+1})-f(x^*))\alpha_k + A_k\frac{f(x_{k+1})-f(x_k)}{\delta} + \varepsilon_k^1\\ &\overset{(2.44b)}{=} -D_f(x^*, x_{k+1})\alpha_k + \varepsilon_k^2\end{aligned}$$

The inequality on the last line follows from the convexity of f and the strict convexity of h. Both errors, $\varepsilon_k^1 = -\frac{1}{\delta}D_h(z_{k+1}, z_k)$ and $\varepsilon_k^2 = -\frac{A_k}{\delta}D_f(x_k, x_{k+1}) - \frac{1}{\delta}D_h(z_{k+1}, z_k)$ are negative. This argument allows us to conclude a $O(1/A_k)$ convergence rate.

Accelerated gradient family. We study families of algorithms which give rise to a family of accelerated methods. These methods can be thought of variations of the explicit Euler scheme applied to (2.41a) and the implicit Euler scheme applied to (2.41b). Take, $\tau_k = (A_{k+1}-A_k)/\delta A_{k+1} := \alpha_k/A_{k+1}$. The variational conditions for the first family of methods can be written as the following:

$$\frac{x_{k+1}-y_k}{\delta} = \tau_k(z_k-y_k) \tag{2.46a}$$

$$\frac{\nabla h(z_{k+1})-\nabla h(z_k)}{\delta} = -\frac{A_{k+1}-A_k}{\delta}\nabla f(x_{k+1}) \tag{2.46b}$$

$$y_{k+1} = \mathcal{G}(x), \tag{2.46c}$$

where $\mathcal{G}: \mathcal{X} \to \mathcal{X}$ is an arbitrary map whose domain is the previous state, $x = (x_{k+1}, z_{k+1}, y_k)$. The variational conditions for second family can be written:

$$\frac{x_{k+1}-y_k}{\delta} = \tau_k(z_k-y_k) \tag{2.47a}$$

$$y_{k+1} = \mathcal{G}(x) \tag{2.47b}$$

$$\frac{\nabla h(z_{k+1})-\nabla h(z_k)}{\delta} = -\frac{A_{k+1}-A_k}{\delta}\nabla f(y_{k+1}), \tag{2.47c}$$

where $\mathcal{G}: \mathcal{X} \to \mathcal{X}$ is an arbitrary map whose domain is the previous state, $x = (x_{k+1}, z_k, y_k)$. When $\mathcal{G}(x) = x_{k+1}$ for either algorithm, we recover a classical explicit discretization applied to (2.41a) and implicit discretization applied to (2.41b). We will show that the additional sequence y_k allows us to obtain better error bounds in our Lyapunov analysis. Indeed, accelerated gradient descent [43, 45], accelerated mirror prox [13] accelerated higher-order methods [40, 6], accelerated universal methods [21], all involve particular choices for the map $\mathcal{G}$ and for the smoothness assumptions on f and h. We demonstrate how the analyses contained in all of these papers implicitly show the following discrete-time Lyapunov

function,

$$E_k = D_h(x^*, z_k) + A_k(f(y_k) - f(x^*)), \tag{2.48}$$

is decreasing for each iteration k. We present the proposition for gradient descent in the main text, and leave the fully general case to the appendix.

Proposition 2.2.3. *Assume that the distance-generating function h is σ-strongly convex and the objective function f is convex. Using only the updates (2.46a) and (2.46b), and using the Lyapunov function (2.48), we have the following bound:*

$$\frac{E_{k+1} - E_k}{\delta} \leq -D_f(x^*, x_{k+1})\alpha_k + \varepsilon_{k+1}, \tag{2.49}$$

where the error term scales as

$$\varepsilon_{k+1} = \delta\frac{\alpha_k^2}{2\sigma}\|\nabla f(y_{k+1})\|^2 + A_{k+1}\frac{f(y_{k+1}) - f(x_{k+1})}{\delta}. \tag{2.50a}$$

If we use the updates (2.47a) and (2.47c) instead, the error term scales as

$$\varepsilon_{k+1} = \delta\frac{\alpha_k^2}{2\sigma}\|\nabla f(y_{k+1})\|^2 + A_{k+1}\left\langle \nabla f(y_{k+1}), \frac{y_{k+1} - x_{k+1}}{\delta}\right\rangle. \tag{2.50b}$$

In particular, accelerated mirror decent uses the following family of operators $\mathcal{G} \equiv \mathcal{G}_\epsilon$, parameterized by a scaling constant $\epsilon > 0$:

$$\mathcal{G}_\epsilon(x) = \arg\min_{y\in\mathcal{X}}\left\{f(x) + \langle\nabla f(x), y - x\rangle + \frac{1}{2\epsilon}\|y - x\|^2\right\}. \tag{2.51}$$

Nesterov assumed the use of full gradients ∇f which are $(1/\epsilon)$-smooth; thus, the gradient map is scaled according to the Lipschitz parameter. Using the gradient update, $y_{k+1} = \mathcal{G}_\epsilon(x_{k+1})$, for updates (2.46c) and (2.47b), where $\mathcal{G}_\epsilon$ is defined in (2.51), the error for algorithm (2.46) can be written as follows:

$$\varepsilon_{k+1} = \delta\frac{\alpha_k^2}{2\sigma}\|\nabla f(x_{k+1})\|^2 - \frac{A_{k+1}\epsilon}{2\delta}\|\nabla f(x_{k+1})\|^2, \tag{2.52a}$$

and for algorithm (2.47), we have:

$$\varepsilon_{k+1} = \delta\frac{\alpha_k^2}{2\sigma}\|\nabla f(y_{k+1})\|^2 - \frac{A_{k+1}\epsilon}{2\delta}\|\nabla f(y_{k+1})\|^2. \tag{2.52b}$$

In particular, observe that the optimality condition for the gradient update (2.51) is

$$\nabla f(x) = \frac{1}{\epsilon}(x - \mathcal{G}_\epsilon(x)). \tag{2.53}$$

The bound (2.52a) follows from smoothness of the objective function f,

$$f(\mathcal{G}_\epsilon(x)) \leq f(x) + \langle \nabla f(x), \mathcal{G}_\epsilon(x) - x\rangle + \frac{\epsilon}{2}\|\mathcal{G}_\epsilon(x) - x\|^2 \overset{(2.53)}{=} f(x) - \frac{\epsilon}{2}\|\nabla f(x)\|^2.$$

For the second bound (2.52b), we use the L-smoothness of the gradient,

$$\|\nabla f(\mathcal{G}_\epsilon(x)) - \nabla f(x)\| \leq \frac{1}{\epsilon}\|\mathcal{G}_\epsilon(x) - x\|; \tag{2.54}$$

substituting (2.53) into (2.54), squaring both sides, and expanding the square on the left-hand side, yields the desired bound:

$$\langle \nabla f(\mathcal{G}_\epsilon(x)), x - \mathcal{G}_\epsilon(x))\rangle \leq -\frac{\epsilon}{2}\|\nabla f(\mathcal{G}_\epsilon(x))\|^2.$$

The error bounds we have just obtained depend explicitly on the scaling ϵ. This restricts our choice of sequences A_k; they must satisfy the following inequality, $\frac{\alpha_k^2}{A_{k+1}} \leq 1$, for the error to be bounded. For example, $A_{k+1} = \frac{\sigma\epsilon(k+1)(k+2)}{4}$ and $\alpha_k = \frac{\sqrt{\sigma\epsilon}(k+1)}{2}$ satisfies the bound; from this we can conclude $f(y_k) - f(x^*) \leq O(1/\epsilon\sigma k^2)$, which matches the lower bound for algorithms which only use full gradients of the objective function. Furthermore, if we take the discretization step to scale according to the smoothness as $\delta = \sqrt{\epsilon\sigma}$, then both $\|\frac{x_k - y_k}{\sqrt{\epsilon}}\| = O(\sqrt{\epsilon})$ and $\varepsilon_k = O(\sqrt{\epsilon})$; therefore, as $\delta = \sqrt{\epsilon\sigma} \to 0$, for a fixed σ, we recover the dynamics (2.41) and the statement $\frac{d}{dt}\mathcal{E}_t \leq 0$ for Lyapunov function (2.38) in the limit.

Accelerated Mirror Prox Accelerated mirror-prox, which was introduced by Diakonikolas and Orecchia [13], also fits into the Lyapunov framework. Let f be $(1/\epsilon)$-smooth. Take, $\tau_k = (A_{k+1} - A_k)/\delta A_{k+1} := \alpha_k/A_{k+1}$. The variational conditions for the first family of methods can be written as the following:

$$\frac{x'_{k+1} - x_k}{\delta} = \tau_k(z_k - x_k) \tag{2.55a}$$

$$\frac{\nabla h(z'_{k+1}) - \nabla h(z_k)}{\delta} = -\frac{A_{k+1} - A_k}{\delta}\nabla f(x'_{k+1}) \tag{2.55b}$$

$$\frac{x_{k+1} - x_k}{\delta} = \tau_k(z'_{k+1} - x_k) \tag{2.55c}$$

$$\frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta} = -\frac{A_{k+1} - A_k}{\delta}\nabla f(x_{k+1}) \tag{2.55d}$$

As an algorithm, we can write it as, $x'_{k+1} = \delta\tau_k z_k + (1 - \delta\tau_k)x_k$ ((2.55a)); $y_{k+1} = \nabla h(z_k) - \alpha_k\nabla f(x'_{k+1})$, $z'_{k+1} = \Pi_{\mathcal{X}\cap\mathcal{X}'}(\nabla h^*(y_{k+1}))$ ((2.55b)); $x_{k+1} = \delta\tau_k z'_{k+1} + (1 - \delta\tau_k)x_k$ ((2.55c)); $y'_{k+1} = \nabla h(z_k) - \alpha_k\nabla f(x_{k+1})$, $z_{k+1} = \Pi_{\mathcal{X}\cap\mathcal{X}'}(\nabla h^*(y'_{k+1}))$, where Π is the Bregman projection operator (B.18) . The function (2.45) is a Lyapunov function for (2.55). In particular, the following upper bound,

$$\frac{E_{k+1} - E_k}{\delta} \leq -D_f(x^*, x_{k+1})\alpha_k + \varepsilon_{k+1},$$

follows using a few simple arguments, where the error scales as,

$$\varepsilon_{k+1} = \delta \frac{\alpha_k^2}{A_{k+1}\epsilon} \|z'_{k+1} - z_k\| \|z'_{k+1} - z_{k+1}\| - \frac{\sigma}{2\delta}\|z'_{k+1} - z_k\|^2 - \frac{\sigma}{2\delta}\|z'_{k+1} - z_{k+1}\|^2. \tag{2.56}$$

The proof of this result can be found in Appendix B.4. Taking $\delta = \sqrt{\epsilon\sigma}$, the error is nonpositive if $\frac{\alpha_k^2}{A_{k+1}} \leq 1$. The same choices, $A_{k+1} = \frac{\sigma\epsilon(k+1)(k+2)}{4}$ and $\alpha_k = \frac{\sqrt{\sigma\epsilon}(k+1)}{2}$ ensures the error is nonpositive; from this we can conclude $f(x_k) - f(x^*) \leq O(1/\epsilon\sigma k^2)$, which matches the lower bound for algorithms which only use full gradients of the objective function. Similar to the mirror prox algorithm, $\|z'_{k+1} - z'_{k+1}\| \leq \sigma^{-1}\|\nabla h(z'_{k+1}) - \nabla h(z_{k+1})\| = \delta\sigma^{-1}\|\alpha_k(\nabla f(x'_{k+1}) - \nabla f(x_{k+1}))\| = O(\delta/\sigma)$, so that in the limit $\delta \to 0$, we recover the continuous-time dynamic (2.41).

2.2.1.3 Methods arising from the second Euler-Lagrange equation

Assume f is μ-strongly convex with respect to h (A.7), h is strictly convex, and the second ideal scaling condition (2.37b) holds with equality. We write the Euler Lagrange equation (2.40) as,

$$\frac{d}{dt}X_t = \frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}}(Z_t - X_t), \tag{2.57a}$$

$$\frac{d}{dt}\nabla h(Z_t) = \frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}}\left(\nabla h(X_t) - \nabla h(Z_t) - \frac{1}{\mu}\nabla f(X_t)\right). \tag{2.57b}$$

For this dynamical system,

$$\mathcal{E}_t = e^{\beta_t}\left(\mu D_h(x^*, Z_t) + f(X_t) - f(x^*)\right) \tag{2.58}$$

is a Lyapunov function. We check,

$$\begin{aligned}
\frac{d}{dt}\mathcal{E}_t &= (\mu D_h(x^*, Z_t) + f(X_t) - f(x^*))\frac{d}{dt}e^{\beta_t} + e^{\beta_t}\frac{d}{dt}f(X_t) - e^{\beta_t}\mu\left\langle \frac{d}{dt}\nabla h(Z_t), x^* - Z_t\right\rangle \\
&\overset{(2.57b)}{=} (\mu D_h(x^*, Z_t) + f(X_t) - f(x^*) + \langle \nabla f(X_t), x^* - Z_t\rangle)\frac{d}{dt}e^{\beta_t} \\
&\quad + e^{\beta_t}\frac{d}{dt}f(X_t) + \mu\langle \nabla h(X_t) - \nabla h(Z_t), x^* - Z_t\rangle \frac{d}{dt}e^{\beta_t} \\
&\overset{(2.57a)}{\leq} (-D_f(x^*, X_t) + \mu D_h(x^*, X_t))\frac{d}{dt}e^{\beta_t} \leq 0.
\end{aligned} \tag{2.59}$$

In third line, we use the Bregman three point identity (A.27) and the inequality follows from the μ-strong convexity of f with respect to h (A.7).

Backward-Backward-Euler Written as an algorithm, the implicit Euler scheme applied to (2.57a) and (2.57b) results in the following updates:

$$z_{k+1} = \underset{\substack{z\in\mathcal{X} \\ x=\frac{\delta\tau_k}{1+\delta\tau_k}z+\frac{1}{1+\delta\tau_k}x_k}}{\arg\min} \left\{ f(x) + \mu D_h(z,x) + \frac{\mu}{\delta\tau_k} D_h\left(z, z_k\right) \right\}, \tag{2.60a}$$

$$x_{k+1} = \frac{\delta\tau_k}{1+\delta\tau_k} z_{k+1} + \frac{1}{1+\delta\tau_k} x_k. \tag{2.60b}$$

Using the following discrete-time Lyapunov function:

$$E_k = A_k(\mu D_h(x^*, z_k) + f(x_k) - f(x^*)), \tag{2.61}$$

we obtain the bound $\frac{E_{k+1}-E_k}{\delta} \leq 0$ for algorithm (2.44). This allows us to conclude a general $O(1/A_k)$ convergence rate for the implicit scheme (2.44). Indeed, the backward-Euler (1.5) discretization applied to dynamics (2.60) satisfies the variational conditions

$$\frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta} = \tau_k\Big(\nabla h(x_{k+1}) - \nabla h(z_{k+1}) - \frac{1}{\mu}\nabla f(x_{k+1})\Big),$$

and

$$\frac{x_{k+1} - x_k}{\delta} = \tau_k(z_{k+1} - x_{k+1}),$$

where $\tau_k = \frac{A_{k+1}-A_k}{\delta A_k}$. Using these variational inequalities, we have the following argument:

$$\begin{aligned}
\frac{E_{k+1} - E_k}{\delta} &= (\mu D_h(x^*, z_{k+1}) + f(x_{k+1}) - f(x^*))\alpha_k + A_k\frac{f(x_{k+1}) - f(x_k)}{\delta} \\
&\quad - A_k\mu\left\langle \frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta}, x^* - z_{k+1}\right\rangle + \varepsilon_k^1 \\
&\overset{(2.60a)}{=} (\mu D_h(x^*, z_{k+1}) + f(x_{k+1}) - f(x^*) + \langle \nabla f(x_{k+1}), x^* - z_{k+1}\rangle)\alpha_k \\
&\quad + A_k\frac{f(x_{k+1}) - f(x_k)}{\delta} + \mu\left\langle \frac{\nabla h(x_{k+1}) - \nabla h(z_{k+1})}{\delta}, x^* - z_{k+1}\right\rangle \alpha_k + \varepsilon_k^1 \\
&\overset{(2.60b)}{=} (-D_f(x^*, x_{k+1}) + \mu D_h(x^*, x_{k+1}))\alpha_k + \varepsilon_k^2 \leq 0.
\end{aligned}$$

The third line follows from the Bregman three-point identity (A.27) and the last line follows from the μ-strong convexity of f with respect to h (A.7). The first error scales as $\varepsilon_k^1 = -\frac{A_k\mu}{\delta} D_h(z_{k+1}, z_k)$ and $\varepsilon_k^2 = -\frac{A_k}{\delta} D_f(x_k, x_{k+1}) - \alpha_k\mu D_h(x_{k+1}, z_{k+1}) + \varepsilon_k^1$. We now focus on analyzing the accelerated gradient family, which can be viewed as a discretization that contains easier subproblems.

Accelerated gradient descent We study a family of algorithms which can be thought of as slight variations of the implicit Euler scheme applied to (2.57a) and the explicit Euler scheme applied to (2.57b)

$$\frac{x_k - y_k}{\delta} = \tau_k(z_k - y_k) \tag{2.62a}$$

$$\frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta} = \tau_k \left(\nabla h(x_k) - \nabla h(z_k) - \frac{1}{\mu} \nabla f(x_k) \right) \tag{2.62b}$$

$$y_{k+1} = \mathcal{G}(x), \tag{2.62c}$$

where $x = (x_k, z_{k+1}, y_k)$ is the previous state and $\tau_k = \frac{A_{k+1} - A_k}{\delta A_{k+1}} := \frac{\alpha_k}{A_{k+1}}$. (2.62a) written as an update, is simply, $x_k = \frac{\delta \tau_k}{1+\delta\tau_k} z_k + \frac{1}{1+\delta\tau_k} y_k$. Note that when $\mathcal{G}(x) = x_k$, we recover classical discretizations. The additional sequence $y_{k+1} = \mathcal{G}(x)$, however, allows us to obtain better error bounds using the Lyapunov analysis. To analyze the general algorithm (2.62), we use the following Lyapunov function:

$$E_k = A_k(\mu D_h(x^*, z_k) + f(y_k) - f(x^*)). \tag{2.63}$$

We begin with the following proposition, which provides an initial error bound for algorithm (2.62) using the general update (2.62c).

Proposition 2.2.4. *Assume the objective function f is μ-uniformly convex with respect to h (A.5) and h is σ-strongly convex. In addition, assume f is $(1/\epsilon)$-smooth. Using the sequences (2.62a) and (2.62b), the following bound holds:*

$$\frac{E_{k+1} - E_k}{\delta} \leq (-D_f(x^*, x_k) + \mu D_h(x^*, x_k))\alpha_k + \varepsilon_k, \tag{2.64}$$

where the error term has the following form:

$$\varepsilon_k = A_{k+1} \frac{f(y_{k+1}) - f(x_k)}{\delta} + A_{k+1} \frac{\sigma\mu}{2\delta} \|\nabla h(z_{k+1}) - \nabla h(z_k)\|^2 - A_{k+1} \frac{\sigma\mu}{2\delta} \|x_k - y_k\|^2$$
$$\alpha_k \left(\langle \nabla f(x_k), y_k - x_k \rangle + (1/\epsilon) D_h(y_k, x_k) - \mu D_h(x_k, z_k) \right)$$

When h is Euclidean, the error simplifies to the following form

$$\varepsilon_{k+1} = A_{k+1} \left(\frac{f(y_{k+1}) - f(x_k)}{\delta} + \delta \frac{\tau_k^2}{2\mu} \|\nabla f(x_k)\|^2 + \delta \left(\frac{\tau_k}{2\epsilon} - \frac{\mu}{2\tau_k} \right) \left\| \frac{x_k - y_k}{\delta} \right\|^2 \right).$$

We present a proof of Proposition 2.2.4 in Appendix B.6.3. The result for accelerated gradient descent can be summed up in the following corollary, which is a consequence of Proposition 2.2.4.

Corollary 2.2.5. *Using the gradient step,*

$$\mathcal{G}(x) = x_k - \epsilon \nabla f(x_k),$$

for update (2.62c) results in an error which scales as

$$\varepsilon_{k+1} = A_{k+1}\left(\frac{\delta\tau_k^2}{2\mu} - \frac{\epsilon}{2\delta}\right)\|\nabla f(x_k)\|^2 + \delta A_{k+1}\left(\frac{\tau_k}{2\epsilon} - \frac{\mu}{2\tau_k}\right)\left\|\frac{x_k - y_k}{\delta}\right\|^2,$$

when h is Euclidean.

The parameter choice $\tau_k = \sqrt{\mu}$, $\delta = \sqrt{\epsilon}$ so that $\delta\tau_k = 1/\sqrt{\kappa}$ ensures the error is non-positive. With this choice, we obtain a linear $O(e^{-\sqrt{\mu\delta}k}) = O(e^{-k/\sqrt{\kappa}})$ convergence rate. In addition, $\|\frac{x_k - y_k}{\sqrt{\delta}}\| = O(\sqrt{\delta})$ and $\varepsilon_k = O(\sqrt{\delta})$, so we recover the dynamical system (2.57) in the Euclidean setting and the continuous Lyapunov argument $\dot{\mathcal{E}}_t \leq 0$ in the limit $\sqrt{\epsilon} = \delta \to 0$.

In Appendix B.6, we provide an analysis of the algorithms that arise from dynamics (2.38) and (2.40) for the following additional two settings:

- f has (ϵ, ν)-Hölder continuous gradients (A.15)
- accelerated higher order gradient methods, such as the accelerated cubic-regularized Newton method [40].

2.2.2 Quasi-monotone methods

For both dynamics (2.41) and (2.57), the full gradients $\nabla f(X_t)$ can be replaced by directional subgradients $G_f(X_t, \dot{X}_t)$, and the same functions (2.42) and (2.58) are Lyapunov functions. However, these Lyapunov functions are not necessarily differentiable in this setting. To adapt the analysis, we follow the technique of Su, Boyd and Candes [70] and summarize with the following theorem:

Theorem 2.2.6. *Take $X(0) = x_0, \dot{X}(0) = 0$ and $\beta_t = p \log t$ for $p > 0$. Given a convex function f with directional subgradient $G_f(x, v)$, assume that*

$$\dot{X}_t = \frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}}(Z_t - X_t) \tag{2.65a}$$

$$\frac{d}{dt}\nabla h(Z_t) = G_f(X_t, \dot{X}_t)\frac{d}{dt}e^{\beta_t}, \tag{2.65b}$$

admits a solution $X(t)$ on $[0, \alpha)$ for some $\alpha > 0$. Then for any $0 < t < \alpha$, $\mathcal{E}_t$ given by (2.42) is a Lyapunov function on $[0, \alpha)$. Given a μ-strongly convex function f with directional subgradients, assume that

$$\dot{X}_t = \frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}}(Z_t - X_t) \tag{2.66a}$$

$$\frac{d}{dt}\nabla h(Z_t) = \frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}}\left(\nabla h(X_t) - \nabla h(Z_t) - \frac{1}{\mu}G_f(X_t, \dot{X}_t)\right) \tag{2.66b}$$

QM Dynamics 1:	$\frac{d}{dt}\nabla h(Z_t) = -G_f(X_t, \dot{X}_t)\frac{d}{dt}e^{\beta_t}$, $G_f(X_t, \dot{X}_t) \in \partial f(X_t)$	$\frac{d}{dt}X_t = \frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}}(Z_t - X_t)$
Function Class	Lyapunov Function	Convergence Rate
Convex	$\mathcal{E}_t = D_h(x^*, X_t) + e^{\beta_t}(f(X_t) - f(x^*))$	$f(X_t) - f(x^*) \leq \frac{\mathcal{E}_{t_0}}{e^{\beta_t}}$
QM Method 1:	$\frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta} = -\alpha_k g(x_{k+1})$, $g(x_{k+1}) \in \partial f(x_{k+1})$	$\frac{x_{k+1}-x_k}{\delta} = \tau_k(z_k - x_{k+1})$
Function Class	Lyapunov Function	Convergence Rate
Convex f is Lipschitz; h σ-strongly convex	$E_k = D_h(x^*, z_k) + A_k(f(x_k) - f(x^*))$	$f(x_k) - f(x^*) \leq \frac{E_0 + \delta\sum_{s=0}^k \varepsilon_s^1}{A_k}$
QM Dynamics 2:	$\frac{d}{dt}\nabla h(Z_t) = (\nabla h(X_t) - \nabla h(Z_t) - \frac{1}{\mu}G_f(X_t, \dot{X}_t))\frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}}$, $G_f(X_t, \dot{X}_t) \in \partial f(X_t)$	$\frac{d}{dt}X_t = \frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}}(Z_t - X_t)$
Function Class	Lyapunov Function	Convergence Rate
f is μ-*uniformly convex w.r.t* h	$\mathcal{E}_t = e^{\beta_t}\left(\mu D_h(x^*, Z_t) + f(X_t) - f(x^*)\right)$	$f(X_t) - f(x^*) \leq O(1/e^{\beta_t})$
QM Method 2:	$\frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta} = -\tau_k(\nabla h(x_{k+1}) - \nabla h(z_{k+1}) - \frac{1}{\mu}g(x_{k+1}))$ $g(x_{k+1}) \in \partial f(x_{k+1})$	$\frac{x_{k+1}-x_k}{\delta} = \tau_k(z_k - x_{k+1})$
Function Class	Lyapunov Function	Convergence Rate
μ-Strongly Convex f is Lipschitz;, h σ-strongly convex	$E_k = A_k(D_h(x^*, z_k) + f(x_k) - f(x^*))$	$f(x_k) - f(x^*) \leq \frac{E_0 + \delta\sum_{s=0}^k \varepsilon_s^2}{A_k}$

Table 2.7: Lyapunov functions for the quasi-monotone (QM) subgradient dynamics and quasi-monotone (QM) subgradient methods. There is a discretization error as we move to discrete time, and we choose parameters accordingly. Here, $e^{\beta_t} = A_k$, so that $\frac{d}{dt}e^{\beta_t} \approx (A_{k+1} - A_k)/\delta = \alpha_k$ and $\tau_k = (A_{k+1} - A_k)/\delta A_k$. The errors scales as $\varepsilon_k^1 = \delta\frac{\alpha_k^2}{2\sigma}G^2$ and $\varepsilon_k^2 = \delta\frac{1}{2\sigma\mu}\frac{\alpha_k^2}{A_k}G^2$. In the limit $\delta \to 0$, the discrete-time and continuous-time statements match.

admits a solution $X(t)$ on $[0, \alpha)$ for some $\alpha > 0$. Then for any $0 < t < \alpha$, $\mathcal{E}_t$ given by (2.58) is a Lyapunov function on $[0, \alpha)$.

The proof can be found in Appendix B.6.4. Notice, this theorem does not guarantee the existence of solutions for (2.65) and (2.66). Let $\tau_k = \frac{A_{k+1}-A_k}{\delta A_k} := \frac{\alpha_k}{A_k}$ and $g(x) \in \partial f(x)$. We analyze the following discretizations

$$\frac{x_{k+1} - x_k}{\delta} = \tau_k(z_k - x_{k+1}) \tag{2.67a}$$

$$\frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta} = -\frac{A_{k+1} - A_k}{\delta}g(x_{k+1}), \tag{2.67b}$$

and,

$$\frac{x_{k+1} - x_k}{\delta} = \tau_k(z_k - x_{k+1}) \tag{2.68a}$$

$$\frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta} = \tau_k \left(\nabla h(x_{k+1}) - \nabla h(z_{k+1}) - \frac{1}{\mu} g(x_{k+1}) \right), \tag{2.68b}$$

using Lyapunov functions (2.45) and (2.61), respectively. When h is Euclidean, we can write (2.68b) as the following update:

$$z_{k+1} = \arg\min_{z \in \mathcal{X}} \left\{ \langle g(x_{k+1}), z \rangle + \frac{\mu}{2\delta\tau_k} \|z - \tilde{z}_{k+1}\|^2 \right\}.$$

where $\tilde{z}_{k+1} = \frac{z_k + \delta\tau_k x_{k+1}}{1+\delta\tau_k}$. The update (2.68b) involves optimizing a linear approximation to the function regularized by a weighted combination of Bregman divergences. The Lyapunov arguments resemble the continuous-time arguments (2.43) and (2.59), respectively. For algorithm (2.67), we use Lyapunov function (2.45) to check,

$$\begin{aligned}
\frac{E_{k+1} - E_k}{\delta} &= \left\langle \frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta}, z_k - x^* \right\rangle + (f(x_{k+1}) - f(x^*))\alpha_k + A_k \frac{f(x_{k+1}) - f(x_k)}{\delta} + \varepsilon_k^1 \\
&\overset{(2.67b)}{=} (\langle g(x_{k+1}), x^* - z_k \rangle + f(x_{k+1}) - f(x^*))\alpha_k + A_k \frac{f(x_{k+1}) - f(x_k)}{\delta} + \varepsilon_k^1 \\
&\overset{(2.67a)}{=} -D_f^g(x^*, x_{k+1})\alpha_k + \varepsilon_k^2.
\end{aligned}$$

Here, the first error scales as $\varepsilon_k^1 = \alpha_k \langle g(x_{k+1}), z_k - z_{k+1} \rangle - \frac{1}{\delta} D_h(z_{k+1}, z_k)$, and the second as $\varepsilon_k^2 = \varepsilon_k^1 - A_k/\delta D_f^g(x_k, x_{k+1})$. The σ-strong convexity of h, Young's inequality, and the Lipschitz property of f ensures the upper bound $\varepsilon_k^2 \leq \delta \frac{\alpha_k^2}{2\sigma} \|g(x_k)\|^2 \leq \delta \frac{\alpha_k^2}{2\sigma} G^2 := \varepsilon_{k+1}$. This allows us to conclude the upper bound

$$f(x_k) - f(x^*) \leq \frac{E_0 + \delta^2 \sum_{s=0}^k \frac{\alpha_s^2}{2\sigma} G^2}{A_k};$$

this bound is the same as the bound obtained for subgradient descent (2.26), but it is on the iterate x_k, and not the time-averaged iterate $\hat{x}_k$. It is maximized with the choice $\alpha_K = D_h(x^*, X_0)/G^2/\sqrt{K}$ which results in an $O(1/\sqrt{K})$ rate of convergence.

For algorithm (2.68), we check

$$\begin{aligned}\frac{E_{k+1}-E_k}{\delta} &= \alpha_k(\mu D_h(x^*,z_{k+1})+f(x_{k+1})-f(x^*))+A_k\frac{f(x_{k+1})-f(x_k)}{\delta}\\ &\quad - A_k\mu\left\langle\frac{\nabla h(z_{k+1})-\nabla h(z_k)}{\delta},x^*-z_{k+1}\right\rangle+\varepsilon_k^1\\ &\overset{(2.60a)}{=} (\mu D_h(x^*,z_{k+1})+f(x_{k+1})-f(x^*)+\langle g(x_{k+1}),x^*-z_k\rangle)\alpha_k\\ &\quad + A_k\frac{f(x_{k+1})-f(x_k)}{\delta}+\mu\langle\nabla h(x_{k+1})-\nabla h(z_{k+1}),x^*-z_{k+1}\rangle\alpha_k+\varepsilon_k^2\\ &\overset{\substack{(A.27)\\(2.60b)}}{=} (-D_f^g(x^*,x_{k+1})+\mu D_h(x^*,x_{k+1}))\alpha_k+\varepsilon_k^3\le 0.\end{aligned}$$

Here, the first error scales as $\varepsilon_k^1=-\frac{A_k\mu}{\delta}D_h(z_{k+1},z_k)$, the second scales as $\varepsilon_k^2=\alpha_k\langle g(x_{k+1}),z_k-z_{k+1}\rangle+\varepsilon_k^1$ and the third as $\varepsilon_k^3=\varepsilon_k^2-\frac{A_k}{\delta}D_f^g(x_k,x_{k+1})\le\varepsilon_k^2$. The σ-strong convexity of h, Young's inequality, and the Lipschitz property of f ensures the upper bound $\varepsilon_k^2\le\delta\frac{\alpha_k^2}{2\mu A_k\sigma}\|g(x_k)\|^2\le\delta\frac{\alpha_k^2}{2\mu A_k\sigma}G^2$. This allows us to conclude the upper bound

$$f(x_k)-f(x^*)\le\frac{E_0+\delta^2\sum_{s=0}^k\frac{\alpha_s^2}{2\mu\sigma A_s}G^2}{A_k};$$

this bound is the same as the bound obtained for the mirror subgradient method (2.24) (it is optimal), but here the convergence rate is on the iterate x_k, and not the time-averaged iterate $\hat{x}_k$. It is maximized by the sequence $A_k=(k+1)k$, which results in the convergence rate $O(1/k)$ convergence rate.

2.2.3 Equivalence between estimate sequences and Lyapunov functions

In this section, we connect our Lyapunov framework directly to estimate sequences. We derive continuous-time estimate sequences directly from our Lyapunov function and demonstrate how these two techniques are equivalent.

Estimate sequences We provide a brief review of the technique of estimate sequences [43]. We begin with the following definition.

Definition 2.2.7. *[43, p. 2.2.1] A pair of sequences $\{\phi_k(x)\}_{k=1}^\infty$ and $\{A_k\}_{k=0}^\infty$ $A_k\ge 1$ is called an* estimate sequence *of function $f(x)$ if*

$$A_k^{-1}\to 0,$$

and, for any $x\in\mathbb{R}^n$ and for all $k\ge 0$, we have

$$\phi_k(x)\le\left(1-A_k^{-1}\right)f(x)+A_k^{-1}\phi_0(x). \tag{2.69}$$

The following lemma, due to Nesterov, explains why estimate sequences are useful.

Lemma 2.2.8. *[43, p. 2.2.1] If for some sequence $\{x_k\}_{k\geq 0}$ we have*

$$f(x_k) \leq \phi_k^* \equiv \min_{x\in\mathcal{X}} \phi_k(x), \tag{2.70}$$

then $f(x_k) - f(x^*) \leq A_k^{-1}[\phi_0(x^*) - f(x^*)]$.

The proof is straightforward:

$$f(x_k) \overset{(2.70)}{\leq} \phi_k^* \equiv \min_{x\in\mathcal{X}} \phi_k(x) \overset{(2.69)}{\leq} \min_{x\in\mathcal{X}} \left[\left(1 - A_k^{-1}\right) f(x) + A_k^{-1}\phi_0(x)\right] \leq \left(1 - A_k^{-1}\right) f(x^*) + A_k^{-1}\phi_0(x^*).$$

Rearranging gives the desired inequality. Notice that this definition is not constructive. Finding sequences which satisfy these conditions is a non-trivial task. The next proposition, formalized by Baes in [6] as an extension of Nesterov's Lemma 2.2.2 [43], provides guidance for constructing estimate sequences. This construction is used in [43, 45, 40, 6, 47, 41], and is, to the best of our knowledge, the only known formal way to construct an estimate sequence. We will see below that this particular class of estimate sequences can be turned into our Lyapunov functions with a few algebraic manipulations (and vice versa).

Proposition 2.2.9. *[6, p. 2.2] Let $\phi_0 : \mathcal{X} \to \mathbb{R}$ be a convex function such that $\min_{x\in\mathcal{X}} \phi_0(x) \geq f^*$. Suppose also that we have a sequence $\{f_k\}_{k\geq 0}$ of functions from $\mathcal{X}$ to $\mathbb{R}$ that underestimates f:*

$$f_k(x) \leq f(x) \quad \text{for all } x \in \mathcal{X} \text{ and all } k \geq 0. \tag{2.71}$$

Define recursively $A_0 = 1$, $\tau_k = \frac{A_{k+1}-A_k}{A_{k+1}} := \frac{\alpha_k}{A_k}$, and

$$\phi_{k+1}(x) := (1-\tau_k)\phi_k(x) + \tau_k f_k(x) = A_{k+1}^{-1}\left(A_0\phi_0(x) + \sum_{s=0}^{k} a_s f_s(x)\right), \tag{2.72}$$

for all $k \geq 0$. Then $(\{\phi_k\}_{k\geq 0}, \{A_k\}_{k\geq 0})$ is an estimate sequence.

From (2.70) and (2.72), we observe that the following invariant:

$$A_{k+1} f(x_{k+1}) \leq \min_x A_{k+1}\phi_{k+1}(x) = \min_x \sum_{s=0}^{k} \alpha_s f_s(x) + A_0\phi_0(x), \tag{2.73}$$

is maintained. In [47, 41], this technique was extended to incorporate an error term $\{\tilde{\varepsilon}_k\}_{k=1}^{\infty}$,

$$\phi_{k+1}(x) - A_{k+1}^{-1}\tilde{\varepsilon}_{k+1} := (1-\tau_k)\left(\phi_k(x) - A_k^{-1}\tilde{\varepsilon}_k\right) + \tau_k f_k(x) = A_{k+1}^{-1}\left(A_0(\phi_0(x) - \tilde{\varepsilon}_0) + \sum_{s=0}^{k} a_s f_s(x)\right),$$

where $\varepsilon_k \geq 0, \forall k$. Rearranging, we have the following bound:

$$A_{k+1}f(x_{k+1}) \leq \min_x A_{k+1}\phi_{k+1}(x) = \min_x \sum_{s=0}^{k} \alpha_s f_s(x) + A_0\Big(\phi_0(x) - A_0^{-1}\tilde{\varepsilon}_0\Big) + \tilde{\varepsilon}_{k+1}.$$

Notice that an argument analogous to that of Lemma 2.2.8 holds:

$$\begin{aligned} A_{k+1}f(x_{k+1}) \leq \sum_{s=0}^{k} \alpha_s f_s(x^*) + A_0(\phi_0(x^*) - \tilde{\varepsilon}_0) + \tilde{\varepsilon}_{k+1} &\overset{(2.71)}{\leq} \sum_{s=0}^{k} \alpha_s f(x^*) + A_0\phi_0(x^*) + \tilde{\varepsilon}_{k+1} \\ &= A_{k+1}f(x^*) + A_0\phi_0(x^*) + \tilde{\varepsilon}_{k+1}. \end{aligned}$$

Rearranging, we obtain the desired bound,

$$f(x_{k+1}) - f(x^*) \leq \frac{A_0\phi_0(x^*) + \tilde{\varepsilon}_{k+1}}{A_{k+1}}.$$

Thus, we simply need to choose our sequences $\{A_k, \phi_k, \tilde{\varepsilon}_k\}_{k=1}^{\infty}$ to ensure $\tilde{\varepsilon}_{k+1}/A_{k+1} \to 0$. The following table illustrates the choices of $\phi_k(x)$ and $\tilde{\varepsilon}_k$ for the four methods discussed earlier.

Algorithm	$f_s(x)$	$\phi_k(x)$	$\tilde{\varepsilon}_{k+1}$
Quasi-Monotone Subgradient Method	linear	$\frac{1}{A_k}D_h(x, z_k) + f(x_k)$	$\frac{1}{2}\sum_{s=0}^{k} \frac{(A_{s+1}-A_s)^2}{2} G^2$
Accelerated Gradient Method (Weakly Convex)	linear	$\frac{1}{A_k}D_h(x, z_k) + f(x_k)$	0
Accelerated Gradient Method (Strongly Convex)	quadratic	$f(x_k) + \frac{\mu}{2}\|x - z_k\|^2$	0
Conditional Gradient Method	linear	$f(x_k)$	$\frac{1}{2\epsilon}\sum_{s=0}^{k} \frac{(A_{s+1}-A_s)^2}{A_{s+1}} diam(\mathcal{X})^2$

Table 2.8: Choices of estimate sequences for various algorithms

In Table 2.8 "linear" is defined as $f_s(x) = f(x_s) + \langle \nabla f(x_s), x - x_s \rangle$, and "quadratic" is defined as $f_s(x) = f(x_s) + \langle \nabla f(x_s), x - x_s \rangle + \frac{\mu}{2}\|x - x_s\|^2$. The estimate-sequence argument is inductive; one must know the three sequences $\{\varepsilon_k, A_k, \phi_k(x)\}$ a priori in order to check the invariants hold. This aspect of the estimate-sequence technique has made it hard to discern its structure and scope.

Equivalence to Lyapunov functions We now demonstrate an equivalence between these two frameworks. The continuous-time view shows that the errors in both the Lyapunov function and estimate sequences are due to discretization errors. We demonstrate how this works for accelerated methods, and defer the proofs for the other algorithms discussed earlier in the chapter to Appendix B.7.

Equivalence in discrete time. The discrete-time estimate sequence (2.72) for accelerated gradient descent can be written:

$$\begin{aligned}\phi_{k+1}(x) &:= f(x_{k+1}) + A_{k+1}^{-1} D_h(x, z_{k+1}) \\ &\overset{(2.72)}{=} (1-\tau_k)\phi_k(x) + \tau_k f_k(x) \\ &\overset{\text{Table 2.8}}{=} \Big(1 - A_{k+1}^{-1}\alpha_k\Big)\Big(f(x_k) + A_k^{-1} D_h(x, z_k)\Big) + A_{k+1}^{-1}\alpha_k f_k(x).\end{aligned}$$

Multiplying through by A_{k+1}, we have the following argument, which follows directly from our definitions:

$$\begin{aligned}A_{k+1} f(x_{k+1}) + D_h(x, z_{k+1}) &= (A_{k+1} - \alpha_k)\Big(f(x_k) + A_k^{-1} D_h(x, z_k)\Big) + \alpha_k f_k(x) \\ &= A_k\Big(f(x_k) + A_k^{-1} D_h(x, z_k)\Big) + (A_{k+1} - A_k) f_k(x) \\ &\leq A_k f(x_k) + D_h(x, z_k) + (A_{k+1} - A_k) f(x).\end{aligned}$$

The last inequality follows from definition (2.71). Rearranging, we obtain the inequality $E_{k+1} \leq E_k$ for our Lyapunov function (2.48). Going the other direction, from our Lyapunov analysis we can derive the following bound:

$$\begin{aligned}E_k &\leq E_0 \\ A_k(f(x_k) - f(x)) + D_h(x, z_k) &\leq A_0(f(x_0) - f(x)) + D_h(x, z_0) \\ A_k\Big(f(x_k) - A_k^{-1} D_h(x, z_k)\Big) &\leq (A_k - A_0) f(x) + A_0\Big(f(x_0) + A_0^{-1} D_h(x^*, z_0)\Big) \\ A_k \phi_k(x) &\leq (A_k - A_0) f(x) + A_0 \phi_0(x). \end{aligned} \tag{2.74}$$

Rearranging, we obtain the estimate sequence (2.69), with $A_0 = 1$:

$$\phi_k(x) \leq \Big(1 - A_k^{-1} A_0\Big) f(x) + A_k^{-1} A_0 \phi_0(x) = \Big(1 - A_k^{-1}\Big) f(x) + A_k^{-1}\phi_0(x).$$

Writing $\mathcal{E}_t \leq \mathcal{E}_0$, one can simply rearrange terms to extract an estimate sequence:

$$f(X_t) + e^{-\beta_t} D_h\left(x, Z_t\right) \leq \Big(1 - e^{-\beta_t} e^{\beta_0}\Big) f(x^*) + e^{-\beta_t} e^{\beta_0}\Big(f(X_0) + e^{-\beta_0} D_h\left(x, Z_0\right)\Big).$$

Comparing this to (2.74), matching terms allows us to extract the continuous-time estimate sequence $\{\phi_t(x), e^{\beta_t}\}$, where $\phi_t(x) = f(X_t) + e^{-\beta_t} D_h(x, Z_t)$.

2.2.4 Dual averaging with momentum

We summarize the results presented in this section in Table 2.9;

DA Dynamic with momentum:	$\frac{d}{dt}Y_t = -\dot{\tau}_t \nabla f(X_t)$, $Y_t = \gamma_t \nabla h(Z_t)$	$\frac{d}{dt}X_t = \frac{\dot{\tau}_t}{\tau_t}(Z_t - X_t)$
Function Class	Lyapunov Function	Convergence Rate
Convex	$\mathcal{E}_t = \gamma_t D_h(x^*, Z_t) + \tau_t(f(X_s) - f(x^*))$	$f(X_t) - f(x^*) \leq \frac{\gamma_t D_h(x^*, X_0)}{\tau_t}$
DA Algorithm with momentum:	$\frac{y_{k+1}-y_k}{\delta} = -\alpha_k g(x_k)$, $y_k = \gamma_k \nabla h(z_k)$	$\frac{x_{k+1}-x_k}{\delta} = \frac{A_{k+1}-A_k}{A_k \delta}(z_k - x_{k+1})$
Function Class	Lyapunov Function	Convergence Rate
Convex f is Lipschitz	$E_k = \gamma_k D_h(x^*, z_k) + A_k(f(x_k) - f(x^*))$	$f(x_k) - f(x^*) \leq \frac{\gamma_k D_h(x^*, z_0) + \delta \sum_{s=0}^{k} \varepsilon_s^1}{A_k}$
Proximal DA Algorithm:	$\frac{y_{k+1}-y_k}{\delta} = -\alpha_k g(x_{k+1})$, $y_k = \gamma_k \nabla h(z_k)$	$\frac{x_{k+1}-x_k}{\delta} = \frac{A_{k+1}-A_k}{A_k \delta}(z_{k+1} - x_{k+1})$
Function Class	Lyapunov Function	Convergence Rate
Convex $\delta > 0$	$E_k = \gamma_k D_h(x^*, x_k) + A_k(f(x_k) - f(x^*))$	$f(x_k) - f(x^*) \leq \frac{\gamma_k D_h(x^*, x_0)}{A_k}$

Table 2.9: Lyapunov functions for the dual averaging dynamic with momentum, dual averaging algorithm with momentum, and the backward-Euler approximation of the dual averaging dynamics with momentum; Here, $g(x) \in \partial f(x)$, $\alpha_k = \frac{A_{k+1}-A_k}{\delta}$, and $\varepsilon_k^1 = \delta \frac{1}{2\sigma} \frac{\alpha_k^2}{\gamma_k} G^2$, where $\|\partial f(x)\|_*^2 \leq G^2$. In the limit $\delta \to 0$, the discrete-time and continuous-time statements match.

We adopt the setting of dual averaging, where we have a pre-establish prox function h with prox-center X_0. When momentum is added to the dual averaging dynamic,

$$\frac{d}{dt}Y_t = -\nabla f(X_t)\frac{d}{dt}\tau_t \tag{2.75a}$$

$$Y_t = \gamma_t \nabla h\left(Z_t\right) \tag{2.75b}$$

$$\frac{d}{dt}X_t = \frac{\dot{\tau}_t}{\tau_t}(Z_t - X_t), \tag{2.75c}$$

the following function,

$$\mathcal{E}_t = \gamma_t D_h(x^*, Z_t) + \tau_t(f(X_t) + f(x^*)), \tag{2.76}$$

is a natural candidate for a Lyapunov function. We check,

$$\begin{aligned}
\frac{d}{dt}\mathcal{E}_t &= D_h(x^*, Z_t)\frac{d}{dt}\gamma_t - \gamma_t \left\langle \frac{d}{dt}\nabla h(Z_t), x^* - Z_t \right\rangle + \dot{\tau}_t(f(X_t) - f(x^*)) + \tau_t \frac{d}{dt} f(X_t) \\
&\overset{(2.75a)}{=} (h(x^*) - h(Z_t))\frac{d}{dt}\gamma_t - \left\langle \frac{d}{dt}Y_t, x^* - Z_t \right\rangle + \dot{\tau}_t(f(X_t) - f(x^*)) + \tau_t \left\langle \nabla f(X_t), \frac{d}{dt}X_t \right\rangle \\
&\overset{(2.75b)}{\overset{(2.75c)}{=}} -\dot{\tau}_t D_f(x^*, X_t) + \dot{\gamma}_t(h(x^*) - h(Z_t)) \leq \dot{\gamma}_t D_h(x^*, Z_0).
\end{aligned}$$

The last inequality uses the fact that $h(x) = D_h(x, Z_0) \geq 0$, $\forall x \in \mathcal{X}$ as well as the definition of a prox-center $h(x^*) = D_h(x^*, Z_0)$. From the bound $\mathcal{E}_t \leq \mathcal{E}_0 + \gamma_t D_h(x^*, Z_0) - \gamma_0 D_h(x^*, Z_0)$,

we obtain the convergence rate[2]

$$f(X_t) - f(x^*) \leq \frac{\mathcal{E}_0 + \gamma_t D_h(x^*, Z_0)}{\tau_t}$$

A similar guarantee can be obtained by the algorithm obtained from discretizing the dynamics (2.75).

Dual Averaging subgradient method with momentum Make the identifications $\tau_t = A_k$, $\dot{\tau}_t = \alpha_k = \frac{A_{k+1}-A_k}{\delta}$ and let $\tau_k = \frac{A_{k+1}-A_k}{A_k \delta}$. The forward-Euler method (1.6) applied to the updates (2.75a) and the backward-Euler method (1.5) to (2.75c) results in the quasi-monotone method,

$$\frac{x_{k+1} - x_k}{\delta} = \tau_k(z_k - x_{k+1}) \tag{2.77a}$$

$$z_{k+1} \in \arg\min_{z \in \mathcal{X}} \left\{ \sum_{s=0}^{k} \alpha_s \langle g(x_{s+1}), z \rangle + \frac{1}{\gamma_k \delta} D_h(z, z_k) \right\}, \tag{2.77b}$$

where $g(x_{k+1}) \in \partial f(x_{k+1})$. The variational condition for (2.77b) is given by

$$\frac{\gamma_{k+1}\nabla h(z_{k+1}) - \gamma_k \nabla h(z_k)}{\delta} = -\alpha_k g(x_{k+1}).$$

The following function,

$$E_k = \gamma_k D_h(x^*, z_k) + A_k(f(x_k) - f(x^*)),$$

is a Lyapunov function. We check,

$$\begin{aligned}
\frac{E_{k+1} - E_k}{\delta} &= D_h(x^*, x_k)\frac{\gamma_{k+1} - \gamma_k}{\delta} - \gamma_k \left\langle \frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta}, x^* - z_{k+1} \right\rangle \\
&\quad + \alpha_k(f(x_{k+1}) - f(x^*)) + A_k \frac{f(x_{k+1}) - f(x_k)}{\delta} + \varepsilon_k^1 \\
&\overset{(2.77b)}{=} -\alpha_k D_f^g(x^*, x_k) + \frac{\gamma_{k+1} - \gamma_k}{\delta}(h(x^*) - h(x_{k+1})) + \varepsilon_k^3 \leq \frac{\gamma_{k+1} - \gamma_k}{\delta} D_h(x_*, z_0) + \varepsilon_k^3.
\end{aligned}$$

where the first error scales as $\varepsilon_k^1 = -\frac{\gamma_k}{\delta} D_h(z_{k+1}, z_k)$, the second as $\varepsilon_k^2 = \alpha_k \langle g(x_{k+1}), x_{k+1} - z_{k+1} \rangle + \varepsilon_k^1$ and $\varepsilon_k^2 = A_k \frac{f(x_{k+1})-f(x_k)}{\delta} + \varepsilon_k^3$. Using the convexity of f, we can bound the error as follows $\varepsilon_k^3 \leq A_k \langle g(x_{k+1}), \frac{x_{k+1}-x_k}{\delta} \rangle + \varepsilon_k^1 = \alpha_k \langle g(x_{k+1}), z_k - z_{k+1} \rangle - \frac{\gamma_k}{\delta} D_h(z_{k+1}, z_k)$. Using the σ-strong convexity of h and Young's inequality and the assumption that f is Lipschitz, $\|\partial f(x)\|_*^2 \leq G^2$, we obtain the upper bounds $\varepsilon_k^3 \leq \frac{\alpha_k^2 \delta}{2\sigma\gamma_k} G^2$. By summing the Lyapunov

[2] we can also write the numerator of the convergence bound as the smaller quantity, $\tau_0(f(X_0) - f(x^*)) + \gamma_t D_h(x^*, Z_0)$

function we obtain the statement, $E_k \leq E_0 + (\gamma_k - \gamma_0) D_h(x^*, z_0) + \delta \sum_{s=0}^{k} \varepsilon_s^3$, from which we obtain the convergence bound,

$$f(x_k) - f(x^*) \leq \frac{E_0 + \gamma_k D_h(x^*, z_0) + \delta^2 \frac{1}{2\sigma} \sum_{s=0}^{k} \frac{\alpha_s^2}{\gamma_s} G^2}{A_k}.$$

If we assume with out loss of generality $\sigma = 1$, and choose $A_k = k$, $\delta = 1$ and $\gamma_k = \frac{G^2}{D_h(x^*, x_0)} \sqrt{k+1}$, we obtain $O(1/\sqrt{k})$ convergence rate [47]. This bound matches the oracle function lower bound for algorithms designed using only subgradients of convex functions (i.e. is provably optimal). Furthermore, as $\delta \to 0$, the error $\varepsilon_k^2 \to 0$ and we recover the result for the continuous time dynamics.

2.2.5 Accelerated Proximal Gradient Dynamics

We summarize several of the results presented in this section in Table 2.10.

Prox AMD Dynamic 1	$\frac{d}{dt}\nabla h(Z_t) = -(\nabla f_2(X_t) + \nabla f_1(Z_t)) \frac{d}{dt} e^{\beta_t}$	$\frac{d}{dt} X_t = \frac{\frac{d}{dt} e^{\beta_t}}{e^{\beta_t}} (Z_t - X_t)$
Function Class	Lyapunov Function	Convergence Rate
Convex	$\mathcal{E}_t = D_h(x^*, Z_t) + e^{\beta_t}(f(X_t) - f(x^*))$	$f(X_t) - f(x^*) \leq O(1/e^{\beta_t})$
Prox AMD Algorithm 1	$\frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta} = -(\nabla f_2(x_{k+1}) - g(z_{k+1}))\alpha_k$ $g(z) \in \partial f_1(z)$, $\|y_k - x_k\| = O(\delta)$, $\delta = \sqrt{\epsilon\sigma}$	$\frac{x_{k+1} - y_k}{\delta} = \tau_k(z_k - y_k)$ $y_{k+1} = y_k + \delta\tau_k(z_{k+1} - y_k)$
Function Class	Lyapunov Function	Convergence Rate
Convex f is $(1/\epsilon)$-smooth, h is σ-strongly convex	$E_k = D_h(x^*, z_k) + A_k(f(y_k) - f(x^*))$	$f(y_k) - f(x^*) \leq O(1/A_k)$
Prox AMD Dynamic 2	$\frac{d}{dt}\nabla h(Z_t) = \frac{\frac{d}{dt} e^{\beta_t}}{e^{\beta_t}} \left(\nabla h(X_t) - \nabla h(Z_t) - \frac{1}{\mu}(\nabla f_2(X_t) + \nabla f_1(Z_t))\right)$	$\frac{d}{dt} X_t = \frac{\frac{d}{dt} e^{\beta_t}}{e^{\beta_t}} (Z_t - X_t)$
Function Class	Lyapunov Function	Convergence Rate
f is μ-*uniformly convex w.r.t* h	$\mathcal{E}_t = e^{\beta_t}\left(\mu D_h(x^*, Z_t) + f(X_t) - f(x^*)\right)$	$f(X_t) - f(x^*) \leq O(1/e^{\beta_t})$
Prox AMD Algorithm 2	$\frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta} = \tau_k \left(\nabla h(x_k) - \nabla h(z_k) - \frac{1}{\mu}(\nabla f(x_k) + g(z_{k+1}))\right)$ $g(z) \in \partial f_1(z)$, $\|y_{k+1} - x_k\| = O(\delta)$, $\delta = \sqrt{\epsilon}$	$\frac{x_k - y_k}{\delta} = \tau_k(z_k - x_k)$ $y_{k+1} = y_k + \delta\tau_k(z_{k+1} - y_k)$
Function Class	Lyapunov Function	Convergence Rate
f is μ-uniformly convex w.r.t h f is $(1/\epsilon)$-smooth, h is Euclidean	$E_k = A_k(D_h(x^*, z_k) + f(y_k) - f(x^*))$	$f(y_k) - f(x^*) \leq O(1/A_k)$

Table 2.10: Lyapunov functions for proximal accelerated mirror descent (AMD) dynamics, proximal accelerated mirror descent (AMD) algorithms . For proximal AMD algorithm 1 we take $A_{k+1} = \frac{\sigma\epsilon(k+1)(k+2)}{4}$, $\alpha_k = \frac{A_{k+1} - A_k}{\delta} = \frac{\sqrt{\sigma\epsilon}(k+2)}{2}$, $\delta = \sqrt{\epsilon\sigma}$ and for proximal AMD algorithm 2, we take $\tau_k = \frac{A_{k+1} - A_k}{\delta A_{k+1}} = \sqrt{\mu}$, $\delta = \sqrt{\epsilon}$.

Convex Functions Define $f = f_1 + f_2$ and assume f_1, f_2 are convex. For the following dynamics,

$$\frac{d}{dt}X_t = \frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}}(Z_t - X_t) \tag{2.78a}$$

$$\frac{d}{dt}\nabla h(Z_t) = -(\nabla f_2(X_t) + \nabla f_1(Z_t))\frac{d}{dt}e^{\beta_t}, \tag{2.78b}$$

the same function (2.42),

$$\mathcal{E}_t = D_h(x^*, Z_t) + e^{\beta_t}(f(X_t) - f(x^*)),$$

is a Lyapunov function for (2.78).

We check,

$$\begin{aligned}\frac{d}{dt}\mathcal{E}_t &= -\left\langle \frac{d}{dt}\nabla h\left(Z_t\right), x^* - Z_t \right\rangle + (f(X_t) - f(x^*))\frac{d}{dt}e^{\beta_t} + e^{\beta_t}\frac{d}{dt}f(X_t) \\ &\overset{(2.78b)}{=} (-D_{f_2}(x^*, X_t) - D_{f_1}(x^*, Z_t) + f_1(X_t) - f_1(Z_t) + \langle \nabla f_2(X_t), X_t - Z_t\rangle)\frac{d}{dt}e^{\beta_t} \\ &\quad + e^{\beta_t}\frac{d}{dt}f(X_t) \overset{(2.78a)}{\leq} -(D_{f_2}(x^*, X_t) + D_{f_1}(x^*, Z_t))\frac{d}{dt}e^{\beta_t} \leq 0\end{aligned}$$

where the second line follows from the dynamical system (2.78a) and (2.78b), and the inequalities follows from the convexity of f_1 and f_2, where we plug in $e^{\beta_t}\frac{d}{dt}f(X_t) \overset{(2.78a)}{=} \langle \nabla f(X_t), Z_t - X_t\rangle \frac{d}{dt}e^{\beta_t}$. This allows us to conclude an $O(e^{-\beta_t})$ convergence rate for the function value

$$f(X_t) - f(x^*) \leq \frac{\mathcal{E}_0}{e^{\beta_t}}.$$

Proximal AGD The backward-Euler discretization of (2.78b) provides us with a forward-backward mapping (B.8)

$$z_{k+1} = \arg\min_{z\in\mathcal{X}} \left\{ f_1(z) + \langle \nabla f_2(x_{k+1}), z\rangle + \frac{1}{\alpha_k}D_h(z, z_k) \right\}. \tag{2.79}$$

Its variational condition is given by

$$\frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta} = -\frac{A_{k+1} - A_k}{\delta}(g(z_{k+1}) + \nabla f_2(x_{k+1})),$$

where $g(z_{k+1}) \in \partial f_1(z_{k+1})$ is an element of the subgradient. We combine this with the forward-Euler method applied to (2.78a), where we have replaced the x_k with an iterate y_k, where $y_{k+1} = \mathcal{G}(x)$, just as in the general AGD setting,

$$\frac{x_{k+1} - y_k}{\delta} = \tau_k(z_k - y_k). \tag{2.80}$$

Here, $\tau_k = \frac{A_{k+1}-A_k}{\delta A_{k+1}} = \frac{\alpha_k}{A_{k+1}}$. We consider maps such that $\|x_k - y_k\| = O(\delta)$ and $x = (x_{k+1}, z_{k+1}, y_k)$ is the previous state. In particular, we choose

$$\frac{y_{k+1} - y_k}{\delta} = \tau_k(z_{k+1} - y_k). \tag{2.81}$$

For this analysis we will need to assume φ is $(1/\epsilon)$-smooth. Using the same Lyapunov function (2.48) as the one used for AGD,

$$E_k = D_h(x^*, z_k) + A_k(f(y_k) - f(x^*)),$$

we check,

$$\begin{aligned}
\frac{E_{k+1} - E_k}{\delta} &= -\left\langle \frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta}, x^* - z_{k+1} \right\rangle + \frac{A_{k+1} - A_k}{\delta}(f(y_{k+1}) - f(x^*)) \\
&\quad + A_k \frac{f(y_{k+1}) - f(y_k)}{\delta} + \varepsilon_k^1 \\
&= (-D_{f_1}(x^*, x_{k+1}) - D_{f_2}^g(x^*, z_{k+1}) + f_1(y_k) - f_1(z_{k+1}) + \langle \nabla f_2(x_{k+1}), x_{k+1} - z_k \rangle)\alpha_k \\
&\quad + A_k \frac{f(y_{k+1}) - f(y_k)}{\delta} + \varepsilon_k^2 \\
&= -(D_{f_1}(x^*, x_{k+1}) + D_{f_2}^g(x^*, z_{k+1}))\alpha_k + \varepsilon_k^3 \leq \varepsilon_k^3.
\end{aligned}$$

We can combine (2.81) and (2.80) to obtain the identity $\frac{x_{k+1}-y_{k+1}}{\delta} = \tau_k(z_k - z_{k+1})$ and $\alpha_k(x_{k+1} - z_k) = A_k \frac{y_k - x_{k+1}}{\delta}$. These identities will be used to simplify the discretization errors.

Here, the error $\varepsilon_k^1 = \frac{1}{\delta} D_h(z_{k+1}, z_k)$, and $\varepsilon_2^k = \varepsilon_1^k + \alpha_k \langle \nabla f_2(x_{k+1}), z_k - z_{k+1} \rangle + \alpha_k(f_2(y_{k+1}) - f_2(x_{k+1})) = \varepsilon_1^k + A_{k+1}\langle \nabla f_2(x_{k+1}), \frac{x_{k+1}-y_{k+1}}{\delta} \rangle + \alpha_k(f_2(y_{k+1}) - f_2(x_{k+1}))$ using the identity. For the last error, we have $\varepsilon_k^3 = \varepsilon_k^2 + A_k \left\langle \nabla f_2(x_{k+1}), \frac{y_k - x_{k+1}}{\delta} \right\rangle + A_k \frac{f_2(y_{k+1}) - f_2(y_k)}{\delta} + A_{k+1} f_1(y_{k+1}) - A_k f_1(y_k) - \alpha_k f_1(z_{k+1}) \leq \varepsilon_k^2 + A_k \frac{f_2(y_{k+1}) - f_2(x_{k+1})}{\delta} + A_{k+1} f_1(y_{k+1}) - A_k f_1(y_k) - \alpha_k f_1(z_{k+1})$, where the upper bounded follows using convexity. First, we notice the convexity of f_1 gives the identity $A_{k+1} f_1((1 - \delta\tau_k)y_k + \delta\tau_k z_{k+1}) \leq A_{k+1}(1 - \delta\tau_k) f_1(y_k) + A_{k+1}\tau_k f_1(z_{k+1})$ using Jensen's (A.4). Therefore $\varepsilon_k^3 \leq \varepsilon_k^2 + A_k \frac{f_2(y_{k+1}) - f_2(x_{k+1})}{\delta}$. Next, we use the σ-strong convexity to upper bound the error as follows, $\varepsilon_k^3 \leq A_k \frac{f_2(y_{k+1}) - f_2(x_{k+1})}{\delta} + \alpha_k(f_2(y_{k+1}) - f_2(x_{k+1})) - \frac{\sigma}{2\delta}\|z_{k+1} - z_k\|^2 + A_{k+1}\langle \nabla f_2(x_{k+1}), \frac{x_{k+1}-y_{k+1}}{\delta} \rangle$. Using the $(1/\epsilon)$-smoothness of f_2, we obtain the upper bound $\varepsilon_k^3 \leq -\frac{\sigma}{2\delta}\|z_{k+1} - z_k\|^2 + \delta A_{k+1}\frac{1}{2\epsilon}\|\frac{y_{k+1}-x_{k+1}}{\delta}\|^2 = -\left(\frac{\sigma}{2\delta} - \delta A_{k+1}\tau_k^2 \frac{1}{2\epsilon}\right)\|z_{k+1} - z_k\|^2$. Making the same parameter as AGD $\delta = \sqrt{\sigma\epsilon}$ and $A_k = \frac{\sigma\epsilon(k+1)(k+2)}{4}$ $\alpha_k = \frac{\sqrt{\sigma\epsilon}(k+2)}{2}$, we can ensure the error ε_k^3 is nonpositive.

Strongly Convex Functions Define $f = f_1 + f_2$ and assume f_2 is μ-strongly convex and f_1 is convex. For the dynamics

$$\frac{d}{dt}X_t = \frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}}(Z_t - X_t) \tag{2.82a}$$

$$\frac{d}{dt}\nabla h(Z_t) = \frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}}\left(\nabla h(X_t) - \nabla h(Z_t) - (1/\mu)(\nabla f_2(X_t) + \nabla f_1(Z_t))\right), \tag{2.82b}$$

the same function (2.16),

$$\mathcal{E}_t = e^{\beta_t}(\mu D_h(x^*, Z_t) + f(X_t) - f(x^*)),$$

is a Lyapunov function.

We check,

$$\begin{aligned}
\frac{d}{dt}\mathcal{E}_t &= (\mu D_h(x^*, Z_t) + f(X_t) - f(x^*))\frac{d}{dt}e^{\beta_t} - \mu e^{\beta_t}\left\langle \frac{d}{dt}\nabla h(Z_t), x^* - Z_t\right\rangle + e^{\beta_t}\frac{d}{dt}f(X_t) \\
&\overset{(2.82b)}{=} (-D_{f_1}(x^*, Z_t) - D_{f_2}(x^*, X_t) + \mu D_h(x^*, Z_t) - \mu\langle \nabla h(X_t) - \nabla h(Z_t), x^* - Z_t\rangle)\frac{d}{dt}e^{\beta_t} \\
&\quad + (\langle \nabla f_2(X_t), X_t - Z_t\rangle + f_1(X_t) - f_1(Z_t))\frac{d}{dt}e^{\beta_t} + e^{\beta_t}\langle \nabla f(X_t), \dot{X}\rangle \\
&\overset{(A.27)}{=} (-D_{f_1}(x^*, Z_t) - D_{f_2}(x^*, X_t) + \mu D_h(x^*, X_t) - \mu D_h(Z_t, X_t))\frac{d}{dt}e^{\beta_t} \\
&\quad (\langle \nabla f_2(X_t), X_t - Z_t\rangle + f_1(X_t) - f_1(Z_t))\frac{d}{dt}e^{\beta_t} + e^{\beta_t}\langle \nabla f(X_t), \dot{X}\rangle \\
&\overset{(2.82a)}{\leq} (-D_{f_2}(x^*, Z_t) - D_{f_1}(x^*, X_t) + \mu D_h(x^*, X_t))\frac{d}{dt}e^{\beta_t} \leq 0.
\end{aligned}$$

Here, the first equality uses the Bregman three-point identity (A.27). The first inequality follows from the convexity of f_1. The last inequality follows from using the strong convexity of f_2.

Accelerated Proximal Gradient Descent We analyze the setting $h(x) = \frac{1}{2}\|x\|^2$. To discretize the dynamics (2.82b), we split the vector field (2.82b) into two components – $v_1(x, z, t) = \frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}}(\nabla h(X_t) - \nabla h(Z_t) - (1/\mu)\nabla f_2(X_t))$ and $v_2(x, z, t) = -\frac{\frac{d}{dt}e^{\beta_t}}{\mu e^{\beta_t}}\nabla f_1(Z_t)$ and apply the forward-Euler scheme to $v_2(x, z, t)$ and the backward-Euler scheme to $v_1(x, z, t)$, with the same identification, $\frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}} = \frac{A_{k+1} - A_k}{\delta A_{k+1}} = \tau_k$ for both vector fields.[3] This results in the

[3]While using the same identification of $\dot{\beta}_t$ for both vector fields is problematic – since one is being evaluated forward in time and the other backward in time – the error bounds only scale sensibly in the setting where $\dot{\beta}_t \leq \sqrt{\mu}$ is a constant.

algorithm,

$$z_{k+1} = \arg\min_z \left\{ f_1(z) + \langle \nabla f_2(x_k), z\rangle + \frac{\mu}{2\delta\tau_k}\|z - (1-\delta\tau_k)z_k - \delta\tau_k x_k\|^2 \right\} \tag{2.83a}$$

$$y_{k+1} = \mathcal{G}(x) \tag{2.83b}$$

$$\frac{x_{k+1} - y_{k+1}}{\delta} = \tau_{k+1}(z_{k+1} - x_{k+1}), \tag{2.83c}$$

which satisfies the variational condition $\frac{z_{k+1}-z_k}{\delta} = \tau_k \left(x_k - z_k - \frac{1}{\mu}\nabla f_2(x_k) - \frac{1}{\mu} g(z_{k+1}) \right)$, where $g(x) \in \partial f_1(x)$. We can combine this update with the backward-Euler method applied to (2.82a), where we have replaced the iterate x_k with an iterate y_{k+1}, just as in the AGD setting. Using the Lyapunov function

$$E_{k+1} = A_k \left(\frac{\mu}{2}\|x^* - z_k\|^2 + f(y_k) - f(x^*) \right)$$

we check,

$$\begin{aligned}
\frac{E_{k+1} - E_k}{\delta} &= \left(\frac{\mu}{2}\|x^* - z_k\|^2 + f(x_k) - f(x^*) \right) \frac{A_{k+1} - A_k}{\delta} - \mu A_{k+1} \left\langle \frac{z_{k+1} - z_k}{\delta}, x^* - z_{k+1} \right\rangle \\
&\quad + A_{k+1} \frac{f(y_{k+1}) - f(y_k)}{\delta} + \varepsilon_k^1 \\
&= \left(-D_{f_1}^G(x^*, z_{k+1}) - D_{f_2}(x^*, x_k) + \frac{\mu}{2}\|x^* - z_k\|^2 - \mu\langle x_k - z_k, x^* - z_k\rangle \right) \frac{A_{k+1} - A_k}{\delta} \\
&\quad + \left(\langle \nabla f_2(x_k), x_k - z_{k+1}\rangle + f_1(y_k) - f_1(z_{k+1})) + f_2(y_k) - f_2(x_k) \right) \frac{A_{k+1} - A_k}{\delta} \\
&\quad + A_{k+1} \frac{f(y_{k+1}) - f(y_k)}{\delta} + \varepsilon_k^2 \\
&\overset{(A.27)}{=} \left(-D_{f_1}^G(x^*, z_{k+1}) - D_{f_2}(x^*, x_k) + \frac{\mu}{2}\|x^* - x_k\|^2 - \frac{\mu}{2}\|x_k - z_k\|^2 \right) \frac{A_{k+1} - A_k}{\delta} \\
&\quad + \left(\langle \nabla f_2(x_k), x_k - z_k\rangle + f_1(y_k) - f_1(z_{k+1})) + f_2(y_k) - f_2(x_k) \right) \frac{A_{k+1} - A_k}{\delta} \\
&\quad + A_{k+1} \frac{f(y_{k+1}) - f(y_k)}{\delta} + \varepsilon_k^2 \\
&= \left(-D_{f_2}(x^*, z_{k+1}) - D_{f_1}(x^*, x_k) + \frac{\mu}{2}\|x^* - x_k\|^2 \right) \frac{A_{k+1} - A_k}{\delta} + \varepsilon_k^3.
\end{aligned}$$

Here, the errors scale as $\varepsilon_k^1 = -\delta A_{k+1}\frac{\mu}{2}\|\frac{z_{k+1}-z_k}{\delta}\|^2$, $\varepsilon_k^2 = \varepsilon_k^1 + \mu\alpha_k\langle x_k - z_k, z_k - z_{k+1}\rangle$, and $\varepsilon_k^3 = \varepsilon_k^2 + \alpha_k(\langle \nabla f_2(x_k), x_k - z_{k+1}\rangle + f_1(y_k) - f_1(z_{k+1}) + f_2(y_k) - f_2(x_k)) + A_{k+1}\frac{f(y_{k+1})-f(y_k)}{\delta}$. Using the convexity of f_1, we conclude $A_{k+1}f_1(y_{k+1}) - A_k f_1(y_k) + \alpha_k f_1(z_{k+1}) \le 0$. Using the strong convexity and smoothness of f_2, we upper-bound the error by $\varepsilon_k^3 \le \varepsilon_k^2 + \alpha_k(\langle \nabla f_2(x_k), x_k - z_{k+1}\rangle + f_2(y_k) - f_2(x_k)) + A_{k+1}\langle \nabla f_2(x_k), \frac{y_k - y_{k+1}}{\delta}\rangle + \frac{A_{k+1}}{2\epsilon}\frac{1}{\delta}\|x_k - y_{k+1}\|^2 - \frac{A_{k+1}\mu}{2\delta}\|x_k - y_k\|^2$. Take $y_{k+1} = \mathcal{G}(x) = \delta\tau_k z_{k+1} + (1 - \delta\tau_k)y_k$. With this choice, $A_{k+1}\langle \nabla f_2(x_k), \frac{y_k - y_{k+1}}{\delta}\rangle = \alpha_k\langle \nabla f_2(x_k), z_{k+1} -$

$y_k\rangle$. Plugging this in the error, we have $\varepsilon_k^3 \leq \varepsilon_k^2 + \alpha_k(f_2(y_k) - f_2(x_k) + \langle \nabla f_2(x_k), x_k - y_k\rangle) + \frac{\delta A_{k+1}}{2\epsilon}\|\frac{x_k - y_{k+1}}{\delta}\|^2 - \delta\frac{A_{k+1}\mu}{2}\|\frac{x_k - y_k}{\delta}\|^2$. Using convexity, of f_2, we have the final error bound $\varepsilon_k^3 \leq -\delta A_{k+1}\frac{\mu}{2}\|\frac{z_{k+1}-z_k}{\delta}\|^2 + \mu\alpha_k\langle x_k - z_k, z_k - z_{k+1}\rangle + \delta\frac{A_{k+1}}{2\epsilon}\|\frac{x_k - y_{k+1}}{\delta}\|^2 - \delta\frac{A_{k+1}\mu}{2}\|\frac{x_k - y_k}{\delta}\|^2$. The final step involves using the identity $x_k - y_{k+1} = \delta\tau_k(\tau_k(x_k - z_k) - (\frac{z_k - z_{k+1}}{\delta}))$. This allows us to upper bound the error by $\varepsilon_k^3 \leq -\delta(A_{k+1}\frac{\mu}{2}\|\frac{z_{k+1}-z_k}{\delta}\|^2 + \mu A_{k+1}\langle \tau_k(x_k - z_k), \frac{z_k - z_{k+1}}{\delta}\rangle + \frac{A_{k+1}\tau_k^2\delta^2}{2\epsilon}\|\tau_k(x_k - z_k) - (z_k - z_{k+1})\|^2 - \frac{A_{k+1}\mu}{2}\|\tau_k(x_k - z_k)\|^2)$. Taking $\delta = \sqrt{\epsilon}$ and $\tau_k = \sqrt{\mu}$, we can check the error ε_k^3 is non-positive by completing the square.

2.3 Summary

The connection between algorithms and dynamical systems bring immense structure to the techniques used to obtain upper bound in optimization; indeed, it has been the primary inspiration to a growing number of works in optimization [13, 74, 52] which propose new techniques. We provide a few examples of other places where we think it can be used below, as well as summarize the Lyapunov functions we have presented in Table 2.11.

2.3.1 Additional Lyapunov Arguments

There are several other methods which fit into this framework that we did not discuss. We provide a high-level summary of some examples, leaving details to the Appendix, or as future work.

- **Conjugate Gradient Method**: In [24], Karimi and Vavasis showed that the Lyapunov function (2.63) can be used to analyze the conjugate gradient method (CG). In future work, if possible, it would be interesting to develop a dynamcial perspective for CG.

- **Adagrad with Momentum**: The Lyapunov framework described in this thesis can be applied to obtain new analyses of adaptive methods, such as Adagrad [16]. Let $\alpha_k = \frac{A_{k+1} - A_k}{\delta}$ and $g(x) \in \partial f(x)$ be an element of the subdifferential of f at x. Adagrad,

$$\frac{x_{k+1} - x_k}{\delta} = -\alpha_k H_k^{-1} g(x_k),$$

  can be analyzed using the Lyapunov function,

$$E_k = \frac{1}{2}\|x^* - x_k\|_{H_k}^2 + \sum_{s=0}^{k-1}(f(x_s) - f(x^*))\frac{A_{s+1} - A_s}{\delta}\delta.$$

  Here, $\|x\|_{H_k}^2 = \langle x, H_k x\rangle$, $0 \prec H_0$, and $0 \preceq \frac{H_{k+1} - H_k}{\delta}$.[4] From the Lyapunov property, we obtain the upper bound $f(\hat{x}_k) - f(x^*) \leq O(1/\sqrt{k})$. A natural way to add momentum

[4] Typically, we choose $H_k = \left(\sum_{i=1}^k g(x_i) \circ g(x_i)\right)^{1/2}$ or $H_k = \text{diag}\left(\left(\sum_{i=1}^k g(x_i) \circ g(x_i)\right)^{1/2}\right)$, where "$\circ$" denotes the entrywise Hadamard product.

$\mathcal{E}_t = \tau_t(f(X_t) - f(x^*))$	$E_k = A_k(f(x_k) - f(x^*))$	
Dynamic	Algorithm	Problem Class
Gradient Flow	*Gradient Descent*	f is differentiable, $(1/\delta)$-smooth, $\tau_t = e^{2\mu t}$ f satisfies PL condition with parameter μ
Frank Wolfe	*Frank Wolfe*	f is $(1/\delta)$-smooth $\mathcal{X} \subseteq \mathbb{R}^d$ is a convex and compact
$\mathcal{E}_t = \tau_t D_h(x^*, X_t)$	$E_k = A_k D_h(x^*, x_k)$	
Dynamic	Algorithm	Problem Class
Mirror Descent Dynamic	*Mirror Descent*	f is $(1/\delta)$-smooth
Gradient Descent Dynamic	*Gradient Descent*	f is $(1/\delta)$-smooth
$\mathcal{E}_t = D_h(x^*, X_t) + \tau_t(f(X_k) - f(x^*))$	$E_k = D_h(x^*, x_k) + A_k(f(x_k) - f(x^*))$	
Dynamic	Algorithm	Problem Class
Mirror Descent Dynamic	*Mirror Descent*	f is $(1/\delta)$-smooth $\tau_t = t$, $A_k = \delta k$
$\mathcal{E}_t = \gamma_t D_h(x^*, X_t) + c\int_0^t (f(X_s) - f(x^*))d\tau_s$	$E_k = \gamma_k D_h(x^*, x_k) + c\sum_{s=0}^{k-1}(f(x_s) - f(x^*))\frac{A_{s+1}-A_s}{\delta}\delta$	
Dynamic	Algorithm	Problem Class
Mirror Descent Dynamic	*Mirror Descent ,Mirror Prox*	f is Lipschitz, $\gamma_t \equiv \gamma_k \equiv 1, c = 1$
Mirror Descent Dynamic	*Mirror Descent*	f is Lipschitz, μ-strongly convex $\gamma_t \equiv \tau_t, \gamma_k = A_k, c = \frac{1}{\mu}$
Dual Averaging Dynamic	*Dual Averaging Algorithm*	f is Lipschitz, $c = 1$
$\mathcal{E}_t = \gamma_t D_h(x^*, Z_t) + \tau_t(f(X_t) - f(x^*))$	$E_k = \gamma_k D_h(x^*, z_k) + A_k(f(x_k) - f(x^*))$	
Dynamic	Algorithm	Problem Class
Accelerated Gradient Descent Dynamic	*Accelerated Gradient descent/Mirror Prox*	f is $(1/\delta)$-smooth, $\gamma_t \equiv \gamma_k \equiv 1$
Quasi-monotone Subgradient Dynamic	*Quasi-monotone subgradient descent*	f is Lipschitz, $\gamma_t \equiv \gamma_k \equiv 1$
Dual Averaging with Momentum Dynamic	*Dual Averaging with Momentum Dynamic*	f is Lipschitz
$\mathcal{E}_t = \tau_t(\mu D_h(x^*, Z_t) + f(X_t) - f(x^*))$	$E_k = A_k(\mu D_h(x^*, z_k) + f(x_k) - f(x^*))$	
Dynamic	Algorithm	Problem Class
Accelerated Gradient Descent Dynamic	*Accelerated Gradient descent*	f is $(1/\delta)$-smooth f is μ-strongly convex
Accelerated Proximal Dynamic	*Accelerated Proximal descent*	f is $(1/\delta)$-smooth f is μ-strongly convex

Table 2.11: List of Lyapunov Arguments in Optimization presented in this thesis (so far).

to Adagrad is via the following averaging step

$$\frac{x_{k+1} - x_k}{\delta} = \tau_k(z_k - x_{k+1}) \tag{2.84a}$$

$$\frac{z_{k+1} - z_k}{\delta} = -\alpha_k H_k^{-1} g(x_{k+1}), \tag{2.84b}$$

where $\tau_k = \frac{\alpha_k}{A_k} = \frac{A_{k+1}-A_k}{\delta A_k}$. Using the Lyapunov framework, we can analyze algorithm (2.84) using the function

$$E_k = \frac{1}{2}\|x^* - z_k\|_{H_k}^2 + A_k(f(x_k) - f(x^*)). \tag{2.85}$$

A demonstration of this result can be found Appendix B.7.4. This analysis allows us to conclude the bound $f(x_k) - f(x^*) \leq O(1/\sqrt{k})$ for (2.84), which has a matching lower bound. Such a algorithm might be useful if we do not care about the regret, but the.

- **Geodesically Convex Functions** Geodesic spaces are metric spaces $(\mathcal{X}, d)$ where there is a path, called a geodesic, connecting every two points $x, y \in \mathcal{X}$. Geodesic (strong) convexity generalizes the idea of (strong) convexity to functions defined on these more general spaces. The length of the paths between two points x, y is equivalent to the geodesic distance between them $d(x, y)$ up to a small precision parameter ϵ. Zhang and Sra [79] showed that if $\mathcal{X}$ is an Alexandrov space (has sectional curvature bounded from below), then there is a natural generalization of the Bregman three-point identity (A.27) to geodesic spaces. In particular, for any $x_{k+1}, x_k, x \in \mathcal{X}$, we have

$$\frac{d(x, x_{k+1}) - d(x, x_k)}{\delta} = \left\langle \frac{1}{\delta}\log_{x_k}(x_{k+1}), \log_{x_s}(x) \right\rangle + \frac{\zeta(\kappa, d(x_k, x))}{2\delta}\|\log_{x_k}(x_{k+1})\|^2$$

where $\log = \exp^{-1} : \mathcal{X} \to \mathsf{T}_x\mathcal{X}$ is the inverse of the exponential map and $\zeta(\kappa, d(x_k, x)) > 0$ is a curvature dependent quantity [79, Cor. 8]. Subsequently, it is easy to check that

$$E_k = d(x^*, x_k) + \delta k(f(x_k) - f(x^*))$$

is a Lyapunov function for gradient descent $\frac{1}{\delta}\log_{x_k}(x_{k+1}) = -g_k$ when f is a $(1/\delta)$-geodesically smooth function and g_k is the gradient of f at x_k. In fact, the results contained in Tables 2.1 and 2.3 can be adapted to this more general setting. Recently, there has been some work extending the idea of averaging to geodesic spaces [34]. We believe the Lyapunov framework provides a systematic way to extend several families of second-order algorithms to this more general setting.

- **Higher-order gradient methods** In [76], a Lyapunov analysis of higher-order gradient methods,

$$x_{k+1} = \mathcal{G}_{\epsilon,p,\nu,N}(x_k) = \arg\min_{y\in\mathcal{X}}\left\{ f_{p-1}(x_k; y) + \frac{N}{\epsilon\tilde{p}}\|x_k - y\|^{\tilde{p}} \right\}, \tag{2.86}$$

was also presented, where $\tilde{p} = p - 1 + \nu$, $N > 1, p \geq 3$ and $f_{p-1}(x; y)$ is given by (A.1). If the p-th order derivatives of (2.86) are $(\frac{(p-1)}{\delta}, \nu)$-Hölder smooth (A.15), the function $E_k = (f(x_k) - f(x^*))^{-\frac{1}{\tilde{p}-1}}$ provides a $O(1/\delta k^{\tilde{p}-1})$. Its continuous time limit,

$$\dot{X}_t = \arg\min_v \left\{ \langle \nabla f(X_t), v\rangle + \frac{1}{\tilde{p}}\|v\|^{\tilde{p}} \right\} = -\frac{\nabla f(X_t)}{\|\nabla f(X_t)\|_*^{\frac{\tilde{p}-2}{\tilde{p}-1}}} \tag{2.87}$$

can be analyzed using the same function $\mathcal{E}_t = (f(X_t) - f(x^*))^{-\frac{1}{\tilde{p}-1}}$, to obtain a matching convergence rate $O(t^{\tilde{p}-1})$. It would be interesting to analyze the rescaled gradient flow (2.87) in other settings as well.

- **Newton's method** Newton's method is one of the most widely used and studied algorithms in optimization. It would be interesting , if possible, to develop a dynamical perspective on the analysis of this family of algorithms as well. For example, it is well-known that the function

$$\mathcal{E}_t = \frac{1}{2}\|\nabla f(X_t)\|^2$$

is a Lypaunov function for the Newton dynamics $\dot{X}_t = -\nabla^2 f(X_t)^{-1}\nabla f(X_t)$. Developing a dynamical perspective of the *analysis* of Newton's method, if possible, would be a potentially interesting avenue of future work.

Next, we demonstrate how this Lyapunov framework for dynamical perspective can be extended to stochastic differential equations and stochastic algorithms, including stochastic gradient descent, stochastic gradient descent with momentum, stochastic dual averaging, and stochastic dual averaging with momentum.

Chapter 3

Stochastic Differential Equations

In this chapter, we focus on optimization problems (1.1) where the objective function is of the form, $f(x)+\sigma(x)$. Here $\sigma(x) \sim P$ represents some zero mean noise process $\mathbb{E}_P[\sigma(x)] = 0$. Many machine learning and statistical problems are posed as stochastic optimization problems. The algorithms we discuss to solve these problems have access to oracle functions that provide it with stochastic gradients or stochastic subgradients. All of them are simple, and scale well, requiring little memory. In Section 3.1 we focus on algorithms that discretize first-order stochastic differential equations. In Section 3.2, we turn our attention to algorithms that discretize second-order stochastic differential equations. We end the chapter with a discussion of coordinate methods, demonstrating in the work *Breaking locality accelerates Block Gauss-Seidal* [73] how the Lyapunov framework can be helpful for deriving novel algorithms.

3.1 First-order Stochastic Differential Equations

The first-order stochastic differential equations that model mirror descent have been studied by many [59] and [37]. In this section we summarize and add to these works. In particular, we emphasize the Lyapunov analysis of several families of stochastic differential equations and several stochastic discrete time algorithms, and demonstrate how to move between these arguments.

Stochastic Dual Averaging Dynamics The stochastic dual averaging dynamics (2.29) is given by the following Ito stochastic differential equations (SDE) [51]

$$dY_t = -(\nabla f(X_t)dt + \sigma_t dB_t)\dot{\tau}_t, \tag{3.1a}$$

$$X_t = \nabla h^*(Y_t/\gamma_t), \tag{3.1b}$$

where the diffusion term $\sigma_t := \sigma(x,t)$ is bounded, $\|\sigma_t\|_F^2 \leq G^2$, $\forall x \in \mathcal{X}, t \geq 0$, and $B_t \in \mathbb{R}^d$ is a standard Brownian motion. In particular, [37, Lemma A.4] implicitly showed that (2.23)

$$\mathcal{E}_t = \gamma_t D_{h^*}(Y_t/\gamma_t, \nabla h(x^*)) + \int_0^t (f(X_s) - f(x^*))d\tau_s. \tag{3.2}$$

Stochastic MD Dynamics:	$dY_t = -\dot{\tau}_t(\nabla f(X_t)dt + \sigma(X_t,t)dB_t)$	$X_t = \nabla h^*(Y_t)$
Function Class	Lyapunov Function	Convergence Rate
Convex	$\mathcal{E}_t = D_h(x^*, X_t) + \int_0^t (f(X_s) - f(x^*))d\tau_s$	$\mathbb{E}[f(\hat{X}_t)] - f(x^*) \leq \frac{\mathcal{E}_0 + \int_0^t \varepsilon_s^1 ds}{\tau_t}$
μ-Strong Convexity	$\mathcal{E}_t = e^{\mu\tau_t} D_h(x^*, X_t)$	$\mathbb{E}[D_h(x^*, X_t)] \leq \frac{\mathcal{E}_0 + \int_0^t \varepsilon_s^2 ds}{e^{\mu\tau_t}}$
	$\mathcal{E}_t = e^{\mu\tau_t} D_h(x^*, X_t) + \frac{1}{\mu}\int_0^t (f(X_s) - f(x^*))de^{\mu\tau_s}$	$\mathbb{E}[f(\hat{X}_t)] - f(x^*) \leq \frac{\mu\mathcal{E}_0 + \mu\int_0^t \varepsilon_s^2}{e^{\mu\tau_t}}$
Mirror Subgradient Method:	$\frac{y_{k+1}-y_k}{\delta} = -\alpha_k(\nabla f(x_k) + \sigma(x_k))$	$x_k = \nabla h^*(y_k)$, $g(x_k) \in \partial f(x_k)$
Function Class	Lyapunov Function	Convergence Rate
Convex f is Lipschitz; h σ-strongly convex	$E_k = D_h(x^*, x_k) + \sum_{s=0}^{k-1}(f(x_s) - f(x^*))\frac{A_{s+1}-A_s}{\delta}\delta$	$\mathbb{E}[f(\hat{x}_k)] - f(x^*) \leq \frac{E_0 + \delta\sum_{s=0}^k \varepsilon_s^3}{A_k}$
μ-Strong Convexity f is Lipschitz; h σ-strongly convex	$E_k = A_k D_h(x^*, x_k)$	$\mathbb{E}[D_h(x^*, x_k)] \leq \frac{E_0 + \delta\sum_{s=0}^k \varepsilon_s^4}{A_k}$
	$E_k = A_k D_h(x^* x_k) + \frac{1}{\mu}\sum_{s=0}^{k-1}(f(x_s) - f(x^*))\frac{A_{s+1}-A_s}{\delta}\delta$	$\mathbb{E}[f(\hat{x}_k)] - f(x^*) \leq \frac{\mu E_0 + \mu\delta\sum_{s=0}^k \varepsilon_s^4}{A_k}$
Stochastic DA Dynamics:	$dY_t = -\dot{\tau}_t(\nabla f(X_t)dt + \sigma(X_t,t)dB_t)$	$X_t = \nabla h^*(Y_t/\gamma_t)$
Function Class	Lyapunov Function	Convergence Rate
Convex	$\mathcal{E}_t = \gamma_t D_h(x^*, X_t) + \int_0^t (f(X_s) - f(x^*))d\tau_s$	$\mathbb{E}[f(\hat{X}_t)] - f(x^*) \leq \frac{\mathcal{E}_0 + \gamma_t D_h(x^*, X_0) + \int_0^t \varepsilon_s^5 ds}{\tau_t}$
Stochastic DA Algorithm:	$\frac{y_{k+1}-y_k}{\delta} = -\frac{A_{k+1}-A_k}{\delta}(\nabla f(x_k) + \sigma(x_k))$	$x_k = \nabla h^*(y_k/\gamma_k)$
Function Class	Lyapunov Function	Convergence Rate
Convex	$E_k = \gamma_k D_h(x^*, x_k) + \sum_{s=0}^{k-1}(f(x_s) - f(x^*))\frac{A_{s+1}-A_s}{\delta}\delta$	$\mathbb{E}[f(\hat{X}_t)] - f(x^*) \leq \frac{E_0 + \gamma_k D_h(x^*, x_0) + \delta\sum_{s=0}^{k-1} \varepsilon_s^6}{A_k}$

Table 3.1: Lyapunov functions for stochastic mirror descent dynamics and algorithm and stochastic dual averaging dynamics and algorithm. Assume $\sigma \preceq \nabla^2 h$ and $\mathbb{E}[\sigma_t] \leq G$, $\mathbb{E}[\|g(x)\|_*] \leq G$ $\forall x \in \mathcal{X}$ and $t \in \mathbb{R}^+$. When f is convex, $\alpha_k = \frac{A_{k+1}-A_k}{\delta}$ and when f is strongly convex $\alpha_k = \frac{A_{k+1}-A_k}{\delta\mu A_{k+1}}$. Here, $\varepsilon_s^1 = \frac{1}{2\sigma}G^2\dot{\tau}_s^2$, $\varepsilon_s^2 = \frac{1}{2\sigma}G^2\frac{(\frac{d}{dt}e^{\mu\tau_t}|_{t=s})^2}{2\mu^2 e^{\mu\tau_s}}$, $\varepsilon_s^3 = \delta\frac{1}{2\sigma}G^2\frac{(A_{s-1}-A_s)^2}{\delta^2}$, $\varepsilon_s^4 = \delta\frac{1}{2\sigma}G^2\frac{(A_{s+1}-A_s)^2}{\delta^2 2\mu^2 A_{s+1}}$, $\varepsilon_s^5 = \frac{1}{2\sigma}G^2\frac{\dot{\tau}_s^2}{\gamma_s}$ and $\varepsilon_s^6 = \delta\frac{1}{2\sigma}G^2\frac{(A_{s+1}-A_s)^2}{\delta^2\gamma_s}$. The scalings on the error and Ito correction terms match.

is a Lyapunov function, where $d\tau_t = \dot{\tau}_t dt$. Take X_0 to be the prox center of h. Denote $\tilde{Z}_t = Y_t/\gamma_t$ so that $X_t = \nabla h^*(\tilde{Z}_t)$. Using Ito's formula, on the first component $\tilde{\mathcal{E}}_t = \gamma_t D_{h^*}(\tilde{Z}_t, \nabla h(x^*))$ we check,

$$\begin{aligned}
d\tilde{\mathcal{E}}_t &= \frac{\partial\tilde{\mathcal{E}}_t}{\partial t}dt + \frac{\partial\tilde{\mathcal{E}}_t}{\partial\tilde{Z}_t}d\tilde{Z}_t + \frac{\dot{\tau}_t^2}{2\gamma_t}\mathrm{tr}(\sigma_t^\top\nabla^2 h^*(\tilde{Z}_t)\sigma_t)dt, \\
&= \dot{\gamma}_t D_{h^*}(\tilde{Z}_t, \nabla h(x^*)) + \gamma_t\langle\nabla h^*(\tilde{Z}_t) - x^*, d\tilde{Z}_t\rangle + \frac{\dot{\tau}_t^2}{2\gamma_t}\mathrm{tr}(\sigma_t^\top\nabla^2 h^*(\tilde{Z}_t)\sigma_t)dt \\
&= \dot{\gamma}_t D_h(x^*, \nabla h^*(\tilde{Z}_t)) + \gamma_t\langle\nabla h^*(\tilde{Z}_t) - x^*, d\tilde{Z}_t\rangle + \frac{\dot{\tau}_t^2}{2\gamma_t}\mathrm{tr}(\sigma_t^\top\nabla^2 h^*(\tilde{Z}_t)\sigma_t)dt.
\end{aligned}$$

With the identity $\gamma_t d\tilde{Z}_t = dY_t - \dot{\gamma}_t \tilde{Z}_t dt$, we proceed,

$$\begin{aligned}
d\tilde{\mathcal{E}}_t &= \dot{\gamma}_t(h(x^*) - h(\nabla h^*(\tilde{Z}_t)))dt + \langle \nabla h^*(\tilde{Z}_t) - x^*, dY_t\rangle + \frac{\dot{\tau}_t^2}{2\gamma_t}\mathrm{tr}(\sigma_t^\top \nabla^2 h^*(\tilde{Z}_t)\sigma_t)dt, \\
&\overset{(3.1b)}{\overset{(3.1a)}{=}} \dot{\gamma}_t(h(x^*) - h(\nabla h^*(\tilde{Z}_t))dt + \dot{\tau}_t\langle x^* - X_t, \nabla f(X_t)dt + \sigma_t dB_t\rangle \\
&\quad + \frac{\dot{\tau}_t^2}{2\gamma_t}\mathrm{tr}(\sigma_t^\top \nabla^2 h^*(\tilde{Z}_t)\sigma_t)dt, \\
&\leq -\dot{\tau}_t(f(X_t) - f(x^*))dt + \dot{\tau}_t\langle \sigma_t dB_t, x^* - X_t\rangle + \dot{\gamma}_t D_h(x^*, X_0)dt + \frac{\dot{\tau}_t^2}{2\gamma_t}\mathrm{tr}(\sigma_t^\top \nabla^2 h^*(\tilde{Z}_t)\sigma_t)dt.
\end{aligned}$$

Here, the inequality follows using the convexity of f and non-negativity of h. The last line uses the prox-center identity $\dot{\gamma}_t h(x^*) = \dot{\gamma}_t D_h(x^*, X_0)$. Integrating, we obtain the bound, Define the time averaged iterate $\hat{X}_t = \int_0^s X_s d\tau_s/\tau_t$. Applying Jensen's $\tau_t f(\hat{X}_t) \leq \int_0^t f(X_s)d\tau_s$ Taking the expectation and integrating the last line, we obtain the following convergence bound,

$$\mathbb{E}[f(\hat{X}_t)] - f(x^*) \leq \frac{\tilde{\mathcal{E}}_0 + \gamma_t D_h(x^*, X_0) + \mathbb{E}[\int_0^t \frac{\dot{\tau}_s^2}{2\gamma_s}\mathrm{tr}(\sigma_s^\top \nabla^2 h^*(\tilde{Z}_t)\sigma_s)ds]}{\tau_t}, \tag{3.3}$$

on the time averaged iterate. Assume $\nabla^2 h^* \preceq \sigma^{-1} I$, or equivalently $\sigma I \preceq \nabla h^2$ and $\|\sigma_t\|_F \leq Gt^q \ \forall x \in \mathcal{X}, t \in \mathbb{R}$. Take $\gamma_t = \sqrt{t}$ and $\tau_t = t$. Then the bound (3.3) implies an $O(t^{-\frac{1}{2}+2q})$ convergence rate. In particular, if $q = 0$ (i.e. the noise is not growing), we obtain a $O(t^{-\frac{1}{2}})$ convergence rate.

Stochastic Dual Averaging The variational condition for the stochastic variant of the dual averaging algorithm (2.31) is given by,

$$\frac{y_{k+1} - y_k}{\delta} = -\frac{A_{k+1} - A_k}{\delta}(\nabla f(x_k) + \sigma(x_k)), \tag{3.4a}$$

$$y_k = \gamma_k \nabla h(x_k), \tag{3.4b}$$

where $\mathbb{E}[\sigma(x)] = 0$. Typically, we write $g(x) = \nabla f(x) + \sigma(x)$, so that $\mathbb{E}[g(x)] = \nabla f(x)$. We can analyze this algorithm using the Lyapunov function

$$E_k = \gamma_k D_h(x^*, x_k) + \sum_{s=0}^{k-1}(f(x_s) - f(x^*))\frac{A_{s+1} - A_s}{\delta}\delta.$$

Note the identity $D_h(x^*, x_k) = D_{h^*}(\nabla h(x_k), \nabla h(x^*))$, which follows from (A.29). We check,

$$\begin{aligned}
\frac{E_{k+1} - E_k}{\delta} &= D_h(x^*, x_{k+1})\frac{\gamma_{k+1} - \gamma_k}{\delta} - \gamma_k \left\langle \frac{\nabla h(x_{k+1}) - \nabla h(x_k)}{\delta}, x^* - x_k \right\rangle \\
&\quad + \alpha_k(f(x_k) - f(x^*)) + \varepsilon_k^1 \\
&= (h(x^*) - h(x_{k+1}))\frac{\gamma_{k+1} - \gamma_k}{\delta} - \left\langle \frac{y_{k+1} - y_k}{\delta}, x^* - x_k \right\rangle + \alpha_k(f(x_k) - f(x^*)) + \varepsilon_k^2 \\
&\overset{(3.4a)}{=} -\alpha_k D_f(x^*, x_k) + \frac{\gamma_{k+1} - \gamma_k}{\delta}(h(x^*) - h(x_{k+1})) + \alpha_k\langle \sigma(x_k), x^* - x_k\rangle + \varepsilon_k^2 \\
&\leq \frac{\gamma_{k+1} - \gamma_k}{\delta} D_h(x_*, x_0) + \alpha_k\langle \sigma(x_k), x^* - x_k\rangle + \varepsilon_k^2.
\end{aligned}$$

Here, the errors scale as $\varepsilon_k^1 = -\frac{\gamma_k}{\delta} D_h(x_{k+1}, x_k)$, and $\varepsilon_k^2 = \alpha_k\langle \nabla f(x_k) + \sigma(x_k), x_k - x_{k+1}\rangle - \frac{\gamma_k}{\delta} D_h(x_{k+1}, x_k)$. The final upper bound follows from noting $-D_f(x^*, x_k) \leq 0$ and using the definition of the prox-center. Denote $g(x) = \nabla f(x) + \sigma(x)$ and assume $\mathbb{E}[\|g(x)\|_*^2] \leq G^2$ for all $x \in \mathcal{X}$ and some constant G. Using the σ-strong convexity of h, we can use Young's inequality to upper bound the error $\mathbb{E}[\varepsilon_k^2] \leq \frac{\alpha_k^2 \delta}{2\sigma\gamma_k} G := \varepsilon_k^3$. Denote $\hat{x}_k = \delta \sum_{s=0}^k x_s \alpha_s / A_k$ as the time-average iterate and note that the inequality $A_k f(\hat{x}_k) \leq \delta \sum_{s=0}^k f(x_s)\alpha_s$ follows from Jensen's (A.4). By summing the Lyapunov function and taking the expectation, we obtain the statement, $A_k(\mathbb{E}[f(\hat{x}_k)] - f(x^*)) \leq \mathbb{E}[E_k] \leq E_0 + \gamma_k D_h(x^*, x_0) - \gamma_0 D_h(x^*, x_0) + \delta \sum_{s=0}^k \mathbb{E}[\varepsilon_i^3]$, from which we obtain the convergence bound,

$$\mathbb{E}[f(\hat{x}_k)] - f(x^*) \leq \frac{E_0 + \gamma_k D_h(x^*, x_0) + \delta^2 \frac{1}{2\sigma} \sum_{s=0}^k \frac{\alpha_s^2}{\gamma_s} G^2}{A_k}.$$

If we take If we assume with out loss of generality $\sigma = \delta = 1$, and choose $A_k = k$ and $\gamma_k = \frac{G^2}{D_h(x^*, x_0)}\sqrt{k+1}$, we obtain $O(1/\sqrt{k})$ convergence rate [44, (2.15)].

3.1.1 Stochastic Mirror Descent

The mirror descent dynamics is given by the following Ito stochastic differential equations (SDE) [51] ,

$$dY_t = -\dot{\tau}_t(\nabla f(X_t)dt + \sigma_t dB_t), \tag{3.5a}$$
$$X_t = \nabla h^*(Y_t). \tag{3.5b}$$

We recognize it as the dual averaging dynamics with $\gamma_t \equiv 1$. In the bound (3.3), if we take $\tau_t = \sqrt{t}$, we obtain a matching $O(t^{\frac{1}{2}-2q})$ convergence rate in the setting when f is convex. Now, we will study the dynamics (3.5) in the setting when f is μ-strongly convex.

3.1.2 Strongly convex functions

When f is μ-strongly convex with respect to h (A.7), [37] implicitly showed that the Lyapunov function,

$$\mathcal{E}_t = e^{\mu\tau_t} D_{h^*}(Y_t, \nabla h(x^*)),$$

can be used to provide a convergence rate for (3.5). Using Ito's formula, we check,

$$\begin{aligned}
d\mathcal{E}_t &= \frac{\partial \mathcal{E}_t}{\partial t}dt + \frac{\partial \mathcal{E}_t}{\partial Y_t}dY_t + \frac{\dot{\tau}_t^2 e^{\mu\tau_t}}{2}\mathrm{tr}(\sigma_t^\top \nabla^2 h^*(Y_t)\sigma_t)dt, \\
&= \dot{\tau}_t e^{\mu\tau_t}\left(\mu D_{h^*}(Y_t, \nabla h(x^*))dt - \langle \nabla h^*(Y_t) - x^*, \nabla f(X_t)dt + \sigma_t dB_t\rangle\right) \\
&\quad + \frac{\dot{\tau}_t^2 e^{\mu\tau_t}}{2}\mathrm{tr}(\sigma_t^\top \nabla^2 h^*(Y_t)\sigma_t)dt \\
&= \dot{\tau}_t e^{\mu\tau_t}\left(\mu D_h(x^*, X_t)dt + \langle \nabla f(X_t), x^* - X_t\rangle dt + \langle \sigma_t dB_t, x^* - X_t\rangle\right) \\
&\quad + \frac{\dot{\tau}_t^2 e^{\mu\tau_t}}{2}\mathrm{tr}(\sigma_t^\top \nabla^2 h^*(Y_t)\sigma_t)dt \\
&\leq -\dot{\tau}_t e^{\mu\tau_t}((f(X_t) - f(x^*))dt + \langle \sigma_t dB_t, x^* - X_t\rangle) + \frac{\dot{\tau}_t^2 e^{\mu\tau_t}}{2}\mathrm{tr}(\sigma_t^\top \nabla^2 h^*(Y_t)\sigma_t)dt.
\end{aligned}$$

The last line follows from the strong convexity assumption. By integrating and taking the expectation, we have the bound,

$$\mathbb{E}[D_{h^*}(Y_t, \nabla h(x^*))] \leq \frac{\mathcal{E}_0 + \mathbb{E}[\int_0^t \frac{\dot{\tau}_s^2 e^{\mu\tau_s}}{2}\mathrm{tr}(\sigma_s^\top \nabla^2 h^*(Y_s)\sigma_s)ds]}{e^{\mu\tau_t}}$$

We can also infer the inequality,

$$\frac{1}{\mu}\mathbb{E}\left[\int_0^t (f(X_s) - f(x^*))de^{\mu\tau_s}\right] + \mathbb{E}[\mathcal{E}_t] - \mathcal{E}_0 - \mathbb{E}\left[\int_0^t \frac{\dot{\tau}_s^2 e^{\mu\tau_s}}{2}\mathrm{tr}(\sigma_s^\top \nabla^2 h^*(Y_s)\sigma_s)ds\right] \leq 0, \quad (3.6)$$

from the argument. Define the average iterate $\hat{X}_t = \int_0^t X_s de^{\mu\tau_s}/e^{\mu\tau_t}$. Using Jensen's, we have the inequality $e^{\mu\tau_t} f(\hat{X}_t) \leq \int_0^t f(X_s)de^{\mu\tau_s}$. Taking the expectation of (3.6), we obtain a convergence bound on the expectation of the optimality gap,

$$\mathbb{E}[f(\hat{X}_t)] - f(x^*) \leq \frac{\mu\mathcal{E}_0 + \int_0^t \frac{\left(\frac{d}{dt}e^{\mu\tau_t}|_{t=s}\right)^2}{2e^{\mu\tau_t}\mu}\mathrm{tr}(\sigma_s^\top \nabla^2 h^*(Y_s)\sigma_s)ds}{e^{\mu\tau_t}}, \quad (3.7)$$

evaluated at the time averaged iterate. Assume $\nabla^2 h^* \preceq \sigma^{-1} I$ (i.e that h is σ-strongly convex), and $\mathbb{E}[\|\sigma_t\|_F^2] \leq Gt^{2q}$. Take $\tau_t = t^p$. Then the bound (3.7) implies an $O(t^{p-3+2q})$ rate of convergence. In particular, if $p = 2$, and $q = 0$, we obtain a $O(t^{-1})$ rate of convergence.

Stochastic Mirror Descent Algorithm Let $\dot{\tau}_t \approx \alpha_k = \frac{A_{k+1}-A_k}{\delta \mu A_{k+1}} \approx \frac{\frac{d}{dt}e^{\mu\tau_t}}{\mu e^{\mu\tau_t}}$. The variational condition for the stochastic mirror descent algorithm given by

$$\frac{\nabla h(x_{k+1}) - \nabla h(x_k)}{\delta} = -\alpha_k(\nabla f(x_k) + \sigma(x_k)), \tag{3.8}$$

where $\mathbb{E}[\sigma(x)] = 0$. We can analyze (3.8) using the Lyapunov function,

$$E_k = A_k D_h(x^*, x_k).$$

We check,

$$\begin{aligned}
\frac{E_{k+1} - E_k}{\delta} &= D_h(x^*, x_k)\frac{A_{k+1} - A_k}{\delta} + A_{k+1}\frac{D_h(x^*, x_{k+1}) - D_h(x^*, x_k)}{\delta} \\
&= A_{k+1}\alpha_k \mu D_h(x^*, x_k) - A_{k+1}\left\langle \frac{\nabla h(x_{k+1}) - \nabla h(x_k)}{\delta}, x^* - x_k \right\rangle + \varepsilon_k^1 \\
&\overset{(3.8)}{=} A_{k+1}\alpha_k(\mu D_h(x^*, x_k) + \langle \nabla f(x_k) + \sigma(x_k), x^* - x_k\rangle) + \varepsilon_k^1 \\
&\leq -A_{k+1}\alpha_k(f(x_k) - f(x^*)) + A_{k+1}\alpha_k\langle x^* - x_k, \sigma(x_k)\rangle + \varepsilon_k^1 \leq \varepsilon_k^1
\end{aligned}$$

where the first error scales as $\varepsilon_k = A_{k+1}(\alpha_k\langle \nabla f(x_k) + \sigma(x_k), x_k - x_{k+1}\rangle - \frac{1}{\delta}D_h(x_{k+1}, x_k)$. The upper bound follows from using the strong convexity of f with respect to h. Denote $g(x) = \nabla f(x). - \sigma(x)$. We can upper bound the final error using the σ strong convexity of h as well as Young's inequality (A.25): $\varepsilon_k \leq \frac{(A_{k+1}-A_k)^2}{2\mu^2\sigma\delta A_{k+1}}\|g(x)\|_*^2$. If we assume $\mathbb{E}[\|g(x)\|_*^2] \leq G^2 \ \forall x \in \mathcal{X}$, then by summing and taking the expectation, we obtain the convergence bound

$$\mathbb{E}[D_h(x^*, x_k)] \leq \frac{E_0 + \delta\frac{1}{2\sigma}\sum_{s=0}^k \frac{(A_{s+1}-A_s)^2}{\mu^2\delta A_{s+1}}G^2}{A_k}.$$

We can also infer,

$$\frac{1}{\mu}\mathbb{E}\left[\sum_{s=0}^{k-1}(f(x_s) - f(x^*))\frac{A_{s+1} - A_s}{\delta}\delta\right] + \mathbb{E}[E_k] - E_0 - \delta\frac{1}{2\sigma}\mathbb{E}\left[\sum_{s=0}^k \frac{(A_{s+1} - A_s)^2}{\mu^2\delta A_{k+1}}G^2\right] \leq 0. \tag{3.9}$$

Define the time-average iterate $x_k = \delta\sum_{s=0}^k x_s(A_{s+1} - A_s)/A_k\delta$. Using Jensen's we have the inequality $A_k f(\hat{x}_k) \leq \sum_{s=0}^t f(x_s)\frac{A_{s+1}-A_s}{\delta}\delta$. Taking the expectation of (3.9), we obtain a convergence bound on the expectation of the optimality gap,

$$\mathbb{E}[f(\hat{x}_k)] - f(x^*) \leq \frac{\mu E_0 + \delta\frac{1}{2\sigma}\sum_{s=0}^k \frac{(A_{s+1}-A_s)^2}{\mu\delta A_{s+1}}G^2}{A_k},$$

evaluated at the time aver iterate. The same parameter choices, $A_k = (k+1)k$ so that $\alpha_k = \frac{2}{\delta\mu(k+2)}$ results in an $O(1/k)$ convergence rate.

SAMD Dynamic 1:	$dY_t = -\dot{\tau}_t(\nabla f(X_t)dt + \sigma(X_t,t)dB_t)$ $Y_t/\gamma_t = \nabla h(Z_t)$	$dX_t = \frac{\dot{\tau}_t}{\tau_t}(\nabla h^*(Y_t/\gamma_t) - X_t)dt$
Function Class	Lyapunov Function	Convergence Rate
Convex	$\mathcal{E}_t = D_{h^*}(Y_t/\gamma_t, \nabla h(x^*)) + \tau_t(f(X_t) - f(x^*))$	$\mathbb{E}[f(\hat{X}_t)] - f(x^*) \leq \frac{\mathcal{E}_0 + \gamma_t D_h(x^*, Z_0) \int_0^t \varepsilon_s^1 ds}{\tau_t}$
SAMD Algorithm 1:	$\frac{y_{k+1}-y_k}{\delta} = -\alpha_k(\nabla f(x_k) + \sigma(x_k))$ $y_k/\gamma_k = \nabla h(z_k)$	$\frac{x_{k+1}-x_k}{\delta} = \frac{A_{k+1}-A_k}{\delta A_k}(\nabla h^*(y_k/\gamma_k) - x_k)$
Function Class	Lyapunov Function	Convergence Rate
Convex	$E_k = D_{h^*}(y_k/\gamma_k, \nabla h(x^*)) + A_k(f(x_k) - f(x^*))$	$\mathbb{E}[f(\hat{x}_k)] - f(x^*) \leq \frac{\mathcal{E}_0 + \gamma_k D_h(x^*, z_0) + \delta \sum_{s=0}^k \varepsilon_s^3}{A_k}$
SAMD Dynamic 2:	$dY_t = \frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}}((\nabla h(X_t) - Y_t)dt - \frac{1}{\mu}(\nabla f(X_t)dt + \sigma(X_t,t)dB_t))$ $Y_t = \nabla h(Z_t)$	$dX_t = \frac{\frac{d}{dt}e^{\beta_t}}{e^{\beta_t}}(\nabla h^*(Y_t) - X_t)dt$
Function Class	Lyapunov Function	Convergence Rate
μ-Strongly Convex	$\mathcal{E}_t = e^{\beta_t}(\mu D_{h^*}(Y_t, \nabla h(x^*)) + f(X_t) - f(x^*))$	$\mathbb{E}[f(\hat{X}_t)] - f(x^*) \leq \frac{\mathcal{E}_0 + \int_0^t \varepsilon_s^2 ds}{e^{\beta_t}}$
SAMD Algorithm 2:	$\frac{y_{k+1}-y_k}{\delta} = \frac{A_{k+1}-A_k}{\delta A_k}((\nabla h(x_k) - y_k) - \frac{1}{\mu}(\nabla f(x_k) + \sigma(x_k)))$ $y_k = \nabla h(z_k)$	$\frac{x_{k+1}-x_k}{\delta} = \frac{A_{k+1}-A_k}{\delta A_k}(\nabla h^*(y_k) - x_k)$
Function Class	Lyapunov Function	Convergence Rate
μ-Strongly Convex	$E_k = A_k(\mu D_{h^*}(y_k, \nabla h(x^*)) + f(x_k) - f(x^*))$	$\mathbb{E}[f(\hat{x}_k)] - f(x^*) \leq \frac{\mathcal{E}_0 + \delta \sum_{s=0}^k \varepsilon_s^4 ds}{A_k}$

Table 3.2: Lyapunov functions for the stochastic accelerated mirror descent (SAMD) dynamics and stochastic mirror descent (SAMD) algorithms. The error in continuous time comes from the Ito correction term. Assume $\sigma \preceq \nabla^2 h$ and $\mathbb{E}[\sigma_t] \leq G$, $\mathbb{E}[\|g(x)\|_*] \leq G$ $\forall x \in \mathcal{X}$ and $t \in \mathbb{R}^+$. Here, $\varepsilon_s^1 = \frac{1}{2\sigma}G^2\frac{\dot{\tau}_s^2}{\gamma_s}$, $\varepsilon_s^2 = \frac{1}{2\sigma}G^2\frac{(A_{s+1}-A_s)^2}{\delta^2\gamma_s}\delta$, $\varepsilon_s^3 = \frac{1}{2\sigma}G^2\frac{\left(\frac{d}{dt}e^{\beta_t}|_{t=s}\right)^2}{2\mu e^{\beta_s}}$, and $\varepsilon_s^4 = \frac{1}{2\sigma}G^2\frac{(A_{s+1}-A_s)^2}{2\delta^2\mu A_s}\delta$. The scalings on the error and Ito correction terms match.

3.2 Second-order Stochastic Differential Equations

Krichene and Bartlett [26] showed the stochastic dual averaging dynamic with momentum (2.75),

$$dY_t = -\dot{\tau}_t(\nabla f(X_t)dt + \sigma(X_t,t)dB_t) \tag{3.10a}$$

$$dX_t = \frac{\dot{\tau}_t}{\tau_t}(\nabla h^*(Y_t/\gamma_t) - X_t)dt \tag{3.10b}$$

can be analyzed using the same Lyapunov function (2.76)

$$\mathcal{E}_t = \gamma_t D_{h^*}(Y_t/\gamma_t, \nabla h(x^*)) + \tau_t(f(X_t) - f(x^*)). \tag{3.11}$$

Here, we take $\dot{\gamma}_t, \gamma_t, \dot{\tau}_t, \tau_t > 0$, $Y_t/\gamma_t = \nabla h(Z_t)$ and using (A.29), we note $D_h(x^*, Z_t) = D_{h^*}(Y_t/\gamma_t, \nabla h(x^*))$. Krichene and Bartlett use Ito's formula

$$d\mathcal{E}_t = \frac{\partial \mathcal{E}_t}{\partial t}dt + \frac{\partial \mathcal{E}_t}{\partial X_t}dX_t + \frac{\partial \mathcal{E}_t}{\partial Y_t}dY_t + \frac{\dot{\tau}_t^2}{2\gamma_t}\mathrm{tr}(\sigma_t^\top \nabla^2 h^*(Z_s)\sigma_t)dt,$$

to show the bound [26, Thm 2],

$$\mathbb{E}[\mathcal{E}_t] = \frac{\mathcal{E}_0 + \gamma_t D_h(x^*, Z_0) + \mathbb{E}[\int_0^t \frac{\dot{\tau}_s^2}{2\gamma_s}\mathrm{tr}(\sigma_s^\top \nabla^2 h^*(Z_s)\sigma_t)ds]}{\tau_t}$$

In particular, if we assume $\nabla^2 h^* \preceq \sigma^{-1} I$ (i.e that h is σ-strongly convex), and $\mathbb{E}\|\sigma_t\|_F \leq G$, then obtain the convergence bound

$$\mathbb{E}[f(X_t)] - f(x^*) \leq \frac{\mathcal{E}_0 + \gamma_t D_h(x^*, Z_0) + \frac{1}{2\sigma}G^2 \int_0^t \frac{\dot{\tau}_s^2}{\gamma_s}ds}{\tau_t} \tag{3.12}$$

More generally, [26] note that if $\|\sigma_t\|_F \leq Gt^q$ and $\tau_t = t^p$, then we can infer the upper bound $O(t^{p-1+2q})$ from convergence rate bound (3.12). If we take $p = 1/2$, then $q < 1/4$ for the bound (3.12) to provide a rate of convergence. If $q = 0$, this bound is also optimized by the choices $\gamma_t = \sqrt{t}$ and $\tau_t = t$, which results in an $O(t^{-1/2})$ convergence rate.

Stochastic Dual Averaging with Momentum We can analyze dual averaging with momentum (2.31) where we replace gradients $\nabla f(x)$ with stochastic estimates of the gradients $\nabla f(x) + \sigma(x) = g(x)$, where $\mathbb{E}[\sigma(x)] = 0$. The variational condition for this algorithm can be written,

$$\frac{x_{k+1} - x_k}{\delta} = \tau_k(z_k - x_{k+1}) \tag{3.13a}$$

$$\frac{y_{k+1} - y_k}{\delta} = -\frac{A_{k+1} - A_k}{\delta}(\nabla f(x_{k+1}) + \sigma(x_{k+1})), \tag{3.13b}$$

where $y_k = \gamma_k \nabla h(z_k)$ and $\tau_k = \frac{A_{k+1} - A_k}{\delta A_k}$. Using the Lyapunov function

$$E_k = \gamma_k D_h(x^*, z_k) + A_k(f(x_k) - f(x^*)),$$

which we can also write as

$$E_k = \gamma_k D_{h^*}(y_k/\gamma_k, \nabla h(x^*)) + A_k(f(x_k) - f(x^*)),$$

similar to (3.11).

We check,

$$\begin{aligned}
\frac{E_{k+1}-E_k}{\delta} &= D_h(x^*,x_k)\frac{\gamma_{k+1}-\gamma_k}{\delta} - \gamma_k\left\langle \frac{\nabla h(z_{k+1})-\nabla h(z_k)}{\delta}, x^*-z_{k+1}\right\rangle \\
&\quad + \alpha_k(f(x_{k+1})-f(x^*)) + A_k\frac{f(x_{k+1})-f(x_k)}{\delta} + \varepsilon_k^1 \\
&\overset{(3.13b)}{=} (h(x^*)-h(x_{k+1}))\frac{\gamma_{k+1}-\gamma_k}{\delta} + \alpha_k\left\langle \nabla f(x_{k+1})+\sigma(x_{k+1}), x^*-x_{k+1}\right\rangle \\
&\quad + \alpha_k(f(x_{k+1})-f(x^*)) + A_k\frac{f(x_{k+1})-f(x_k)}{\delta} + \varepsilon_k^2 \\
&= -\alpha_k D_f(x^*,x_k) + \frac{\gamma_{k+1}-\gamma_k}{\delta}(h(x^*)-h(x_{k+1})) + \langle\sigma(x_{k+1}), x^*-z_k\rangle + \varepsilon_k^3 \\
&\leq \frac{\gamma_{k+1}-\gamma_k}{\delta}D_h(x^*,x_0) + \langle\sigma(x_{k+1}), x^*-z_k\rangle + \varepsilon_k^3.
\end{aligned}$$

where the first error scales as $\varepsilon_k^1 = -\frac{\gamma_k}{\delta}D_h(z_{k+1},z_k)$, the second as $\varepsilon_k^2 = \alpha_k\langle\nabla f(x_{k+1}) + \sigma(x_{k+1}), x_{k+1}-z_{k+1}\rangle + \varepsilon_k^1$ and $\varepsilon_k^2 = A_k\frac{f(x_{k+1})-f(x_k)}{\delta} + \varepsilon_k^3$. Denote $g(x) = \nabla f(x) + \sigma(x)$. Using the convexity of f, we can bound the error as follows $\varepsilon_k^3 \leq \alpha_k\langle g(x_{k+1}), z_k - z_{k+1}\rangle - \frac{\gamma_k}{\delta}D_h(z_{k+1},z_k)$. Using the σ-strong convexity of h and Young's inequality and the assumption, $\mathbb{E}[\|g(x)\|_*^2 \leq G^2\ \forall x \in \mathcal{X}$, we obtain the upper bounds $\varepsilon_k^3 \leq \frac{\alpha_k^2\delta}{2\sigma\gamma_k}G^2$. By summing the Lyapunov function we obtain the statement, $E_{A_k} \leq E_{A_0} + (\gamma_k-\gamma_0)D_h(x^*,z_0) + \delta\sum_{s=0}^k\varepsilon_s^3 + \delta\langle(\sigma(x_{s+1}), x^*-z_s\rangle$. Taking the expectation, we obtain the convergence bound,

$$\mathbb{E}[f(x_k)] - f(x^*) \leq \frac{E_0 + \gamma_k D_h(x^*,z_0) + \delta^2\frac{1}{2\sigma}\sum_{s=0}^k\frac{\alpha_s^2}{\gamma_s}G^2}{A_k}.$$

If we assume with out loss of generality $\sigma = 1$, and choose $A_k = k$, $\delta = 1$ and $\gamma_k = \frac{G^2}{D_h(x^*,x_0)}\sqrt{k+1}$, we obtain $O(1/\sqrt{k})$ convergence rate.

3.2.1 Strongly convex functions

In [4], we proposed the dynamics,

$$dX_t = \dot\beta_t(\nabla h^*(Y_t) - X_t)dt \tag{3.14a}$$

$$dY_t = \dot\beta_t\left((\nabla h(X_t) - Y_t)dt - \frac{1}{\mu}(\nabla f(X_t)dt + \sigma_t dB_t)\right) \tag{3.14b}$$

We recognize (3.14) as (2.57) with the addition of a stochastic term, using the identification $\nabla h^*(Y_t) = Z_t$. To analyze the dynamics (3.14), we use the Lyapunov function (2.58)

$$\mathcal{E}_t = e^{\beta_t}\left(\mu D_{h^*}(Y_t, \nabla h(x^*)) + f(X_t) - f(x^*)\right) \tag{3.15}$$

We use Ito's formula

$$d\mathcal{E}_t = \frac{\partial \mathcal{E}_t}{\partial t}dt + \frac{\partial \mathcal{E}_t}{\partial X_t}dX_t + \frac{\partial \mathcal{E}_t}{\partial Y_t}dY_t + \frac{e^{\beta_t}\dot{\tau}_t^2}{2\mu}\mathrm{tr}(\sigma_t^\top \nabla^2 h^*(Y_s)\sigma_t)dt,$$

where we compute the components,

$$\begin{aligned}
\frac{\partial \mathcal{E}_t}{\partial t}dt &= \dot{\beta}_t e^{\beta_t}\left(\mu D_{h^*}(Y_t, \nabla h(x^*)) + f(X_t) - f(x^*)\right)dt \\
\left\langle \frac{\partial \mathcal{E}_t}{\partial X_t}, dX_t \right\rangle &= \dot{\beta}_t e^{\beta_t}\langle \nabla f(X_t), \nabla h^*(Y_t) - X_t\rangle dt \\
\left\langle \frac{\partial \mathcal{E}_t}{\partial Y_t}, dY_t \right\rangle &= \mu\dot{\beta}_t e^{\beta_t}\langle \nabla h^*(Y_t) - x^*, (\nabla h(X_t) - Y_t)dt - \frac{1}{\mu}\left(\nabla f(X_t)dt + \sigma_t dB_t\right)\rangle.
\end{aligned}$$

Subsequently,

$$\begin{aligned}
d\mathcal{E}_t &= \dot{\beta}_t e^{\beta_t}\left(-D_f(x^*, X_t)dt + \mu\left(D_{h^*}(Y_t, \nabla h(x^*))dt + \langle \nabla h^*(Y_t) - x^*, \nabla h(X_t) - Y_t\rangle\right)dt\right) \\
&\quad - \dot{\beta}_t e^{\beta_t}\langle \nabla h^*(Y_t) - x^*, \sigma_t dB_t\rangle + \frac{e^{\beta_t}\dot{\beta}_t^2}{2\mu}\mathrm{tr}(\sigma_t^\top \nabla^2 h^*(Y_s)\sigma_t)dt \\
&\overset{\text{(A.29)}}{\underset{\text{(B.15)}}{=}} \dot{\beta}_t e^{\beta_t}\left(-D_f(x^*, X_t)dt + \mu\left(D_h(x^*, \nabla h^*(Y_t))dt + \langle \nabla h(\nabla h^*(Y_t)) - \nabla h(X_t), x^* - \nabla h^*(Y_t)\rangle\right)dt\right) \\
&\quad - \dot{\beta}_t e^{\beta_t}\langle \nabla h^*(Y_t) - x^*, \sigma_t dB_t\rangle + \frac{e^{\beta_t}\dot{\beta}_t^2}{2\mu}\mathrm{tr}(\sigma_t^\top \nabla^2 h^*(Y_s)\sigma_t)dt \\
&= \dot{\beta}_t e^{\beta_t}\left(-D_f(x^*, X_t)dt + \mu D_h(x^*, X_t)dt - D_h(\nabla h^*(Y_t), X_t)dt - \langle \nabla h^*(Y_t) - x^*, \sigma_t dB_t\rangle\right) \\
&\quad + \frac{e^{\beta_t}\dot{\beta}_t^2}{2\mu}\mathrm{tr}(\sigma_t^\top \nabla^2 h^*(Y_s)\sigma_t)dt \\
&\leq \dot{\beta}_t e^{\beta_t}\langle \nabla h^*(Y_t) - x^*, \sigma_t dB_t\rangle + \frac{e^{\beta_t}\dot{\beta}_t^2}{2\mu}\mathrm{tr}(\sigma_t^\top \nabla^2 h^*(Y_s)\sigma_t)dt.
\end{aligned}$$

The inequality follows from teh strong conveixy to f with respect to h and non-negativity of the bregman divergence for convex functions. Taking the integral of both sides, we have

$$\mathcal{E}_t \leq \mathcal{E}_0 + \int_0^t \dot{\beta}_s e^{\beta_s}\mu\langle \nabla h^*(Y_s) - x^*, \sigma_s dB_s\rangle + \frac{e^{\beta_s}\dot{\beta}_s^2}{2\mu}\mathrm{tr}(\sigma_s^\top \nabla^2 h^*(Y_s)\sigma_s)ds.$$

Finally, taking the expectation, we have

$$\mathbb{E}[\mathcal{E}_t] \leq \mathcal{E}_0 + \mathbb{E}\left[\int_0^t \frac{\left(\frac{d}{dt}e^{\beta_t}|_{t=s}\right)^2}{2\mu e^{\beta_s}}\mathrm{tr}(\sigma_s^\top \nabla^2 h^*(Y_s)\sigma_s)ds\right],$$

from which we can conclude the convergence bound

$$\mathbb{E}[f(X_t)] - f(x^*) \leq \frac{\mathcal{E}_0 + \mathbb{E}\left[\int_0^t \frac{e^{\beta_s}\dot{\beta}_s^2}{2\mu}\mathrm{tr}(\sigma_s^\top \nabla^2 h^*(Y_s)\sigma_s)ds\right]}{e^{\beta_t}}. \tag{3.16}$$

Assume $\nabla^2 h^* = [\nabla^2 h]^{-1} \preceq \sigma^{-1} I$ and $\|\sigma_t\|_F \leq G t^q$. Take $\beta_t = p \log t$ or $e^{\beta_t} = t^p$. Then we can infer the upper bound $O(t^{p-3+2q})$ from (3.16). Take $p = 2$. Then $q < 1/2$ for the upper bound to provide a rate of convergence, otherwise, if $q = 0$, we can infer a $O(t^{-1})$ rate of convergence.

Stochastic Gradient Descent with Momentum When f is μ-strongly convex with respect to h, and h is σ-strongly convex, the algorithm which satisfies the variational condition

$$\frac{x_{k+1} - x_k}{\delta} = \tau_k (z_k - x_{k+1}) \tag{3.17a}$$

$$\frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta} = \tau_k \left(\nabla h(x_{k+1}) - z_k - \frac{1}{\mu} (\nabla f(x_{k+1}) + \sigma(x_{k+1})) \right) \tag{3.17b}$$

can be analyzed using the following Lyapunov function.

$$E_k = A_k (\mu D_h(x^*, z_k) + f(x_k) - f(x^*)).$$

Here, $\sigma(x) = g(x) - \nabla f(x)$, $\mathbb{E}[\sigma(x)] = 0$ and $\tau_k = \frac{A_{k+1} - A_k}{\delta A_k} := \frac{\alpha_k}{A_k}$. Update (3.17b) involves optimizing a linear approximation to the function regularized by a weighted combination of Bregman divergences. When h is Euclidean, we can write (3.17b) as,

$$z_{k+1} = \arg\min_{z \in \mathcal{X}} \left\{ \langle g(x_{k+1}), z \rangle + \frac{\mu}{2\tau_k} \|z - \tilde{z}_{k+1}\|^2 \right\}.$$

With the identification $\nabla h(Y_t) = Z_t$, is similar to the analysis of the SDE (3.14). We check,

$$\begin{aligned}
\frac{E_{k+1} - E_k}{\delta} &= \frac{A_{k+1} - A_k}{\delta} (\mu D_h(x^*, z_{k+1}) + f(x_{k+1}) - f(x^*)) + A_k \frac{f(x_{k+1}) - f(x_k)}{\delta} \\
&\quad - A_k \mu \left\langle \frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta}, x^* - z_{k+1} \right\rangle + \varepsilon_k^1 \\
&\overset{(3.17a)}{=} (\mu D_h(x^*, z_{k+1}) + f(x_{k+1}) - f(x^*) + \langle \nabla f(x_{k+1}) + \sigma(x_{k+1}), x^* - z_k \rangle) \alpha_k \\
&\quad + A_k \frac{f(x_{k+1}) - f(x_k)}{\delta} + \mu \langle \nabla h(x_{k+1}) - \nabla h(z_{k+1}), x^* - z_{k+1} \rangle \alpha_k + \varepsilon_k^2 \\
&\overset{\substack{(A.27) \\ (3.17b)}}{=} (-D_f^g(x^*, x_{k+1}) + \mu D_h(x^*, x_{k+1}) + \langle \sigma(x_{k+1}), x^* - z_k \rangle) \alpha_k + \varepsilon_k^3 \leq 0.
\end{aligned}$$

Here, the first error scales as $\varepsilon_k^1 = -\frac{A_k \mu}{\delta} D_h(z_{k+1}, z_k)$, the second scales as $\varepsilon_k^2 = \alpha_k \langle \nabla f(x_{k+1}) + \sigma(x_{k+1}), z_k - z_{k+1} \rangle + \varepsilon_k^1$ and the third as $\varepsilon_k^3 = \varepsilon_k^2 - \frac{A_k}{\delta} D_f(x_k, x_{k+1}) \leq \varepsilon_k^2$. The σ-strong convexity of h, Young's inequality, and the assumption $\mathbb{E}[\|g(x)\|] \leq G$, ensures $\varepsilon_k^2 \leq \delta \frac{\alpha_k^2}{2\mu A_k \sigma} \|g(x_k)\|^2 \leq \delta \frac{\alpha_k^2}{2\mu A_k \sigma} G^2$. This allows us to conclude the upper bound

$$\mathbb{E}[f(x_k)] - f(x^*) \leq \frac{E_0 + \delta^2 \sum_{s=0}^k \frac{\alpha_s^2}{2\mu\sigma A_s} G^2}{A_k}.$$

Notice, for all the methods analyzed in this section, the Ito correction term scales in the same way as the term we refer to as the discrete-time error.

3.3 Lyapunov arguments for coordinate methods

Coordinate methods are another class of iterative methods in which only a few components of the state x are updated at any given time. For example, the coordinate version of gradient descent has as its update,

$$\frac{x_{k+1} - x_k}{\delta} = -\nabla_i f(x_k)$$

where ∇_i is the gradient of f along its i-th coordinate, which is sampled randomly $i \in [d]$, so that $\mathbb{E}[\nabla_i f(x)] = \nabla f(x)$. The Lyapunov framework can be extended to analyze coordinate version of the algorithms discussed in the previous chapter. As a preview, we a present a Lyapunov argument for the coordinate version of accelerated mirror prox (2.55), as that does not have appeared to be done yet. We end by presenting our work *Breaking Locality Accelerates Block Gauss Seidel*, where we demonstrate how the Lyapunov framework can be used to analyze coordinate algorithms with very general sampling schemes.

Coordinate Accelerated Mirror Prox Sample coordinate $i \in [d]$ at random. We assume f is convex along its i-th coordinate (A.20) and $(1/\epsilon_i)$ smooth along its i-th coordinate (A.21). The block coordinate mirror prox method can be written as the sequence of updates,

$$\frac{x'_{k+1} - x_k}{\delta} = \tau_k(z_k - x_k) \tag{3.18a}$$

$$z'^{(i)}_{k+1} = \arg\min_{x \in \mathcal{X}} \left\{ \langle \nabla_i f(x'_{k+1}), x \rangle + D_h(x, z_k) \right\} \tag{3.18b}$$

$$\frac{x^{(i)}_{k+1} - x_k}{\delta} = \tau_k(z'^{(i)}_{k+1} - x_k) \tag{3.18c}$$

$$z^{(i)}_{k+1} = \arg\min_{x \in \mathcal{X}} \left\{ \langle \nabla_i f(x^{(i)}_{k+1}), x \rangle + D_h(x, z_k) \right\} \tag{3.18d}$$

We use the superscript $^{(i)}$ on $z'^{(i)}_{k+1}, x^{(i)}_{k+1}$ and $z^{(i)}_{k+1}$ to denote that its value depends on the choice of coordinate i. That is, we perform the update x'_{k+1} and treat x'_{k+1}, z_k and x_k as fixed. We then sample the i-th coordinate along which we compute the relevant gradients and update z'_{k+1}, x_{k+1} and z_{k+1}. Update (3.18b) satisfies the variational condition, $\frac{\nabla h(z'^{(i)}_{k+1}) - \nabla h(z_k)}{\delta} = -\frac{A_{k+1} - A_k}{\delta} \nabla_i f(x'_{k+1})$, and update (3.18b) satisfies the variational condition, $\frac{\nabla h(z^{(i)}_{k+1}) - \nabla h(z_k)}{\delta} =$

$-\frac{A_{k+1}-A_k}{\delta}\nabla_i f(x_{k+1}^{(i)})$. We use the Lyapunov function (2.45) to analyze (3.18). We check,

$$\begin{aligned}\frac{E_{k+1}-E_k}{\delta} &= -\left\langle \frac{\nabla h(z_{k+1}^{(i)})-\nabla h(z_k)}{\delta}, x^*-x_{k+1}\right\rangle + \alpha_k(f(x_{k+1}^{(i)})-f(x^*)) + \varepsilon_k^1 \\ &\overset{(2.55d)}{=} -D_{f^i}(x^*, x_{k+1}^{(i)})\alpha_k + \varepsilon_k^1\end{aligned}$$

where we use the notation $D_{f^i}(x,y)$ is given by (A.19). Here, the error scales as,

$$\begin{aligned}\varepsilon_k^1 &= A_k\frac{f(x_{k+1}^{(i)})-f(x_k)}{\delta} - \left\langle \frac{\nabla h(z_{k+1}^{(i)})-\nabla h(z_k)}{\delta}, x_{k+1}^{(i)}-z_{k+1}\right\rangle - \frac{1}{\delta}D_h(z_{k+1}^{(i)}, z_k) \\ &\overset{\substack{(3.18d)\\(A.27)}}{=} A_k\frac{f(x_{k+1}^{(i)})-f(x_k)}{\delta} + \alpha_k\langle\nabla_i f(x_{k+1}^{(i)}), x_{k+1}^{(i)}-z_{k+1}^{(i)}\rangle - \frac{1}{\delta}D_h(z_{k+1}^{(i)}, z_{k+1}^{\prime(i)}) \\ &\quad - \frac{1}{\delta}D_h(z_{k+1}^{\prime(i)}, z_k) - \left\langle \frac{\nabla h(z_{k+1}^{\prime(i)})-\nabla h(z_k)}{\delta}, z_{k+1}^{(i)}-z_{k+1}^{\prime(i)}\right\rangle\end{aligned}$$

Using convexity along the i-th coordinate, we can further upper-bound the error as follows,

$$\begin{aligned}\varepsilon_k^1 &\overset{(3.18b)}{\leq} A_{k+1}\left\langle \nabla_i f(x_{k+1}^{(i)}), \frac{x_{k+1}^{(i)}-x_k}{\delta}\right\rangle + \alpha_k\langle\nabla_i f(x_{k+1}^{(i)}), x_k-z_{k+1}^{(i)}\rangle - \frac{1}{\delta}D_h(z_{k+1}^{(i)}, z_{k+1}^{\prime(i)}) \\ &\quad - \frac{1}{\delta}D_h(z_{k+1}^{\prime(i)}, z_k) + \alpha_k\langle\nabla_i f(x_{k+1}^{\prime(i)}), z_{k+1}^{(i)}-z_{k+1}^{\prime(i)}\rangle \\ &\overset{(3.18c)}{=} \alpha_k\langle\nabla_i f(x_{k+1}^{(i)})-\nabla_i f(x_{k+1}^{\prime}), z_{k+1}^{\prime(i)}-z_{k+1}^{(i)}\rangle - \frac{1}{\delta}D_h(z_{k+1}^{(i)}, z_{k+1}^{\prime(i)}) - \frac{1}{\delta}D_h(z_{k+1}^{\prime}, z_k)\end{aligned}$$

Using the $(1/\epsilon_i)$-smoothness of $\nabla_i f$, Cauchy-Schwartz (A.26) and the identity $\frac{x_{k+1}^{(i)}-x_{k+1}^{\prime}}{\delta} = \tau_k(z_{k+1}^{\prime(i)}-z_k)$, the inequality $\alpha_k\langle\nabla_i f(x_{k+1}^{(i)})-\nabla_i f(x_{k+1}^{\prime}), z_{k+1}^{\prime(i)}-z_{k+1}^{(i)}\rangle \leq \alpha_k\|\nabla_i f(x_{k+1}^{(i)})-\nabla_i f(x_{k+1}^{\prime})\|\|z_{k+1}^{\prime(i)}-z_{k+1}^{(i)}\| \overset{\substack{(3.18a)\\(3.18c)}}{=} \delta\frac{\alpha_k^2}{A_{k+1}\epsilon_i}\|z_{k+1}^{\prime(i)}-z_{k+1}^{(i)}\|\|z_{k+1}^{\prime(i)}-z_k\|$ and the σ-strong convexity of h gives the upper bound

$$\frac{E_{k+1}-E_k}{\delta} \leq -D_{f^i}(x^*, x_{k+1}^{(i)}) + \varepsilon_k^1$$

where

$$\varepsilon_k^1 = \delta\frac{\alpha_k^2}{A_{k+1}\epsilon_i}\|z_{k+1}^{\prime(i)}-z_k\|\|z_{k+1}^{\prime(i)}-z_{k+1}^{(i)}\| - \frac{\sigma}{2\delta}\|z_{k+1}^{\prime(i)}-z_k\|^2 - \frac{\sigma}{2\delta}\|z_{k+1}^{\prime(i)}-z_{k+1}^{(i)}\|^2.$$

Taking the expectation of of both sides ensures $\frac{\mathbb{E}[E_{k+1}]-E_k}{\delta} \leq \mathbb{E}[\varepsilon_k^1]$. Taking $\delta = \sqrt{\epsilon\sigma}$, the expected error is nonpositive if $\frac{\alpha_k^2}{A_{k+1}} \leq 1$. The same choices as mirror prox/agd, $A_{k+1} = \frac{\sigma\epsilon(k+1)(k+2)}{4}$ and $\alpha_k = \frac{\sqrt{\sigma\epsilon}(k+1)}{2}$ ensures the error is nonpositive; from this we can conclude $\mathbb{E}[f(x_k)] - f(x^*) \leq O(1/\epsilon\sigma k^2)$.

Summary From this example, it is clear that how the aforementioned Lyapunov framework can be applied to the coordinate versions of all the methods previously discussed. We now show how we can use this framework can be extended to analyze coordinate methods which perform random coordinate block updates at every iteration.

3.4 Breaking Locality Accelerates Block Gauss-Seidel

This section is based on the work *Breaking locality accelerates block Gauss-Seidel.* S. Tu, S. Venkataraman, A. Wilson, A. Gittens, M. I. Jordan, and B. Recht. In D. Precup and Y. W. Teh (Eds), Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, Australia, NY, 2017.

Recent work by Nesterov and Stich [48] showed that momentum can be used to accelerate the rate of convergence for block Gauss-Seidel in the setting where a fixed partitioning of the coordinates is chosen ahead of time. We show that this setting is too restrictive, constructing instances where breaking locality by running non-accelerated Gauss-Seidel with randomly sampled coordinates substantially outperforms accelerated Gauss-Seidel with any fixed partitioning. Motivated by this finding, we analyze the accelerated block Gauss-Seidel algorithm in the random coordinate sampling setting. Our Lyapunov framework captures the benefit of acceleration with a new data-dependent parameter which is well behaved when the matrix sub-blocks are well-conditioned. Empirically, we show that accelerated Gauss-Seidel with random coordinate sampling provides speedups for large scale machine learning tasks when compared to non-accelerated Gauss-Seidel and the classical conjugate-gradient

3.4.1 Introduction

The randomized Gauss-Seidel method is a commonly used iterative algorithm to compute the solution of an $n \times n$ linear system $Ax = b$ by updating a single coordinate at a time in a randomized order. While this approach is known to converge linearly to the true solution when A is positive definite (see e.g. [31]), in practice it is often more efficient to update a small block of coordinates at a time due to the effects of cache locality.

In extending randomized Gauss-Seidel to the block setting, a natural question that arises is how one should sample the next block. At one extreme a *fixed partition* of the coordinates is chosen ahead of time. The algorithm is restricted to randomly selecting blocks from this fixed partitioning, thus favoring data locality. At the other extreme we break locality by sampling a new set of *random coordinates* to form a block at every iteration.

Theoretically, the fixed partition case is well understood both for Gauss-Seidel [58, 20] and its Nesterov accelerated variant [48]. More specifically, at most $O(\mu_{\text{part}}^{-1} \log(1/\varepsilon))$ iterations of Gauss-Seidel are sufficient to reach a solution with at most ε error, where μ_{part} is a quantity which measures how well the A matrix is preconditioned by the block diagonal matrix containing the sub-blocks corresponding to the fixed partitioning. When acceleration

is used, Nesterov and Stich [48] show that the rate improves to $O\left(\sqrt{\frac{n}{p}\mu_{\text{part}}^{-1}}\log(1/\varepsilon)\right)$, where p is the partition size.

For the random coordinate selection model, the existing literature is less complete. While it is known [58, 20] that the iteration complexity with random coordinate section is $O(\mu_{\text{rand}}^{-1}\log(1/\varepsilon))$ for an ε error solution, μ_{rand} is another instance dependent quantity which is not directly comparable to μ_{part}. Hence it is not obvious how much better, if at all, one expects random coordinate selection to perform compared to fixed partitioning.

Our first contribution in this paper is to show that, when compared to the random coordinate selection model, the fixed partition model can perform very poorly in terms of iteration complexity to reach a pre-specified error. Specifically, we present a family of instances (similar to the matrices recently studied by Lee and Wright [28]) where *non*-accelerated Gauss-Seidel with random coordinate selection performs *arbitrarily* faster than both non-accelerated and even accelerated Gauss-Seidel, using *any* fixed partition. Our result thus shows the importance of the sampling strategy and that acceleration cannot make up for a poor choice of sampling distribution.

This finding motivates us to further study the benefits of acceleration under the random coordinate selection model. Interestingly, the benefits are more nuanced under this model. We show that acceleration improves the rate from $O(\mu_{\text{rand}}^{-1}\log(1/\varepsilon))$ to $O\left(\sqrt{\nu\mu_{\text{rand}}^{-1}}\log(1/\varepsilon)\right)$, where ν is a new instance dependent quantity that satisfies $\nu \leq \mu_{\text{rand}}^{-1}$. We derive a bound on ν which suggests that if the sub-blocks of A are all well conditioned, then acceleration can provide substantial speedups. We note that this is merely a sufficient condition, and our experiments suggest that our bound is conservative.

In the process of deriving our results, we also develop a general proof framework for randomized accelerated methods based on Wilson et al. [77] which avoids the use of estimate sequences in favor of an explicit Lyapunov function. Using our proof framework we are able to recover recent results [48, 1] on accelerated coordinate descent. Furthermore, our proof framework allows us to immediately transfer our results on Gauss-Seidel over to the randomized accelerated Kaczmarz algorithm, extending a recent result by Liu and Wright [33] on updating a single constraint at a time to the block case.

Finally, we empirically demonstrate that despite its theoretical nuances, accelerated Gauss-Seidel using random coordinate selection can provide significant speedups in practical applications over Gauss-Seidel with fixed partition sampling, as well as the classical conjugate-gradient (CG) algorithm. As an example, for a kernel ridge regression (KRR) task in machine learning on the augmented CIFAR-10 dataset ($n = 250,000$), acceleration with random coordinate sampling performs up to $1.5\times$ faster than acceleration with a fixed partitioning to reach an error tolerance of 10^{-2}, with the gap substantially widening for smaller error tolerances. Furthermore, it performs over $3.5\times$ faster than conjugate-gradient on the same task.

3.4.2 Background

We assume that we are given an $n \times n$ matrix A which is positive definite, and an n dimensional response vector b. We also fix an integer p which denotes a block size. Under the assumption of A being positive definite, the function $f(x) = \frac{1}{2}x^\top Ax - x^\top b$ is strongly convex and smooth. Recent analysis of Gauss-Seidel [20] proceeds by noting the connection between Gauss-Seidel and (block) coordinate descent on f. This is the point of view we will take in this paper.

3.4.2.1 Existing rates for randomized block Gauss-Seidel

We first describe the sketching framework of [58, 20] and show how it yields rates on Gauss-Seidel when blocks are chosen via a fixed partition or randomly at every iteration. While we will only focus on the special case when the sketch matrix represents column sampling, the sketching framework allows us to provide a unified analysis of both cases.

To be more precise, let $\mathcal{D}$ be a distribution over $\mathbb{R}^{n\times p}$, and let $S_k \sim \mathcal{D}$ be drawn iid from $\mathcal{D}$. If we perform block coordinate descent by minimizing f along the range of S_k, then the randomized block Gauss-Seidel update is given by

$$x_{k+1} = x_k - S_k(S_k^\mathsf{T} A S_k)^\dagger S_k^\mathsf{T}(Ax_k - b)\,. \tag{3.19}$$

Column sampling. Every index set $J \subseteq 2^{[n]}$ with $|J| = p$ induces a sketching matrix $S(J) = (e_{J(1)}, ..., e_{J(p)})$ where e_i denotes the i-th standard basis vector in $\mathbb{R}^n$, and $J(1), ..., J(p)$ is any ordering of the elements of J. By equipping different probability measures on $2^{[n]}$, one can easily describe fixed partition sampling as well as random coordinate sampling (and many other sampling schemes). The former puts uniform mass on the index sets $J_1, ..., J_{n/p}$, whereas the latter puts uniform mass on all $\binom{n}{p}$ index sets of size p. Furthermore, in the sketching framework there is no limitation to use a uniform distribution, nor is there any limitation to use a fixed p for every iteration. For this paper, however, we will restrict our attention to these cases.

Existing rates. Under the assumptions stated above, [58, 20] show that for every $k \geq 0$, the sequence (3.19) satisfies

$$\mathbb{E}[\|x_k - x_*\|_A] \leq (1-\mu)^{k/2}\|x_0 - x_*\|_A\,, \tag{3.20}$$

where $\mu = \lambda_{\min}(\mathbb{E}[P_{A^{1/2}S}])$. The expectation in (3.20) is taken with respect to the randomness of $S_0, S_1, ...$, and the expectation in the definition of μ is taken with respect to $S \sim \mathcal{D}$. Under both fixed partitioning and random coordinate selection, $\mu > 0$ is guaranteed (see e.g. [20], Lemma 4.3). Thus, (3.19) achieves a linear rate of convergence to the true solution, with the rate governed by the μ quantity shown above.

We now specialize (3.20) to fixed partitioning and random coordinate sampling, and provide some intuition for why we expect the latter to outperform the former in terms of

iteration complexity. We first consider the case when the sampling distribution corresponds to fixed partitioning. Assume for notational convenience that the fixed partitioning corresponds to placing the first p coordinates in the first partition J_1, the next p coordinates in the second partition J_2, and so on. Here, $\mu = \mu_{\text{part}}$ corresponds to a measure of how close the product of A with the inverse of the block diagonal is to the identity matrix, defined as

$$\mu_{\text{part}} = \frac{p}{n}\lambda_{\min}\left(A \cdot \text{blkdiag}\left(A_{J_1}^{-1}, ..., A_{J_{n/p}}^{-1}\right)\right) . \tag{3.21}$$

Above, A_{J_i} denotes the $p \times p$ matrix corresponding to the sub-matrix of A indexed by the i-th partition. A loose lower bound on μ_{part} is

$$\mu_{\text{part}} \geq \frac{p}{n}\frac{\lambda_{\min}(A)}{\max_{1\leq i\leq n/p}\lambda_{\max}(A_{J_i})} . \tag{3.22}$$

On the other hand, in the random coordinate case, Qu et al. [58] derive a lower bound on $\mu = \mu_{\text{rand}}$ as

$$\mu_{\text{rand}} \geq \frac{p}{n}\left(\beta + (1-\beta)\frac{\max_{1\leq i\leq n} A_{ii}}{\lambda_{\min}(A)}\right)^{-1} , \tag{3.23}$$

where $\beta = (p-1)/(n-1)$. Using the lower bounds (3.22) and (3.23), we can upper bound the iteration complexity of fixed partition Gauss-Seidel N_{part} by $O\left(\frac{n}{p}\frac{\max_{1\leq i\leq n/p}\lambda_{\max}(A_{J_i})}{\lambda_{\min}(A)}\log(1/\varepsilon)\right)$ and random coordinate Gauss-Seidel N_{rand} as $O\left(\frac{n}{p}\frac{\max_{1\leq i\leq n} A_{ii}}{\lambda_{\min}(A)}\log(1/\varepsilon)\right)$. Comparing the bound on N_{part} to the bound on N_{rand}, it is not unreasonable to expect that random coordinate sampling has better iteration complexity than fixed partition sampling in certain cases. In Section 3.4.3, we verify this by constructing instances A such that fixed partition Gauss-Seidel takes arbitrarily more iterations to reach a pre-specified error tolerance compared with random coordinate Gauss-Seidel.

3.4.2.2 Accelerated rates for fixed partition Gauss-Seidel

Based on the interpretation of Gauss-Seidel as block coordinate descent on the function f, we can use Theorem 1 of Nesterov and Stich [48] to recover a procedure and a rate for accelerating (3.19) in the fixed partition case; the specific details are discussed in Section C.4.2 of the appendix. We will refer to this procedure as ACDM.

The convergence guarantee of the ACDM procedure is that for all $k \geq 0$,

$$\mathbb{E}[\|x_k - x_*\|_A] \leq O\left(\left(1 - \sqrt{\frac{p}{n}\mu_{\text{part}}}\right)^{k/2}\|x_0 - x_*\|_A\right) . \tag{3.24}$$

Above, μ_{part} is the same quantity defined in (3.21). Comparing (3.24) to the non-accelerated Gauss-Seidel rate given in (3.20), we see that acceleration improves the iteration complexity to reach a solution with ε error from $O(\mu_{\text{part}}^{-1}\log(1/\varepsilon))$ to $O\left(\sqrt{\frac{n}{p}\mu_{\text{part}}^{-1}}\log(1/\varepsilon)\right)$, as discussed in Section 3.4.1.

3.4.3 Results

We now present the main results. All proofs are deferred to the appendix.

3.4.3.1 Fixed partition vs random coordinate sampling

Our first result is to construct instances where Gauss-Seidel with fixed partition sampling runs arbitrarily slower than random coordinate sampling, even if acceleration is used.

Consider the family of $n \times n$ positive definite matrices $\mathscr{A}$ given by $\mathscr{A} = \{A_{\alpha,\beta} : \alpha > 0, \alpha + \beta > 0\}$ with $A_{\alpha,\beta}$ defined as $A_{\alpha,\beta} = \alpha I + \frac{\beta}{n} 1_n 1_n^\mathsf{T}$. The family $\mathscr{A}$ exhibits a crucial property that $\Pi^\mathsf{T} A_{\alpha,\beta} \Pi = A_{\alpha,\beta}$ for every $n \times n$ permutation matrix Π. Lee and Wright [28] recently exploited this invariance to illustrate the behavior of cyclic versus randomized permutations in coordinate descent.

We explore the behavior of Gauss-Seidel as the matrices $A_{\alpha,\beta}$ become ill-conditioned. To do this, we consider a particular parameterization which holds the minimum eigenvalue equal to one and sends the maximum eigenvalue to infinity via the sub-family $\{A_{1,\beta}\}_{\beta>0}$. Our first proposition characterizes the behavior of Gauss-Seidel with fixed partitions on this sub-family.

Proposition 3.4.1. *Fix $\beta > 0$ and positive integers n, p, k such that $n = pk$. Let $\{J_i\}_{i=1}^k$ be any partition of $\{1, ..., n\}$ with $|J_i| = p$, and denote $S_i \in \mathbb{R}^{n \times p}$ as the column selector for partition J_i. Suppose $S \in \mathbb{R}^{n \times p}$ takes on value S_i with probability $1/k$. For every $A_{1,\beta} \in \mathscr{A}$ we have that*

$$\mu_{\text{part}} = \frac{p}{n + \beta p} \,. \tag{3.25}$$

Next, we perform a similar calculation under the random column sampling model.

Proposition 3.4.2. *Fix $\beta > 0$ and integers n, p such that $1 < p < n$. Suppose each column of $S \in \mathbb{R}^{n \times p}$ is chosen uniformly at random from $\{e_1, ..., e_n\}$ without replacement. For every $A_{1,\beta} \in \mathscr{A}$ we have that*

$$\mu_{\text{rand}} = \frac{p}{n + \beta p} + \frac{(p-1)\beta p}{(n-1)(n + \beta p)} \,. \tag{3.26}$$

The differences between (3.25) and (3.26) are striking. Let us assume that β is much larger than n. Then, we have that $\mu_{\text{part}} \approx 1/\beta$ for (3.25), whereas $\mu_{\text{rand}} \approx \frac{p-1}{n-1}$ for (3.26). That is, μ_{part} can be made arbitrarily smaller than μ_{rand} as β grows.

Our next proposition states that the rate of Gauss-Seidel from (3.20) is tight order-wise in that for any instance there always exists a starting point which saturates the bound.

Proposition 3.4.3. *Let A be an $n \times n$ positive definite matrix, and let S be a random matrix such that $\mu = \lambda_{\min}(\mathbb{E}[P_{A^{1/2}S}]) > 0$. Let x_* denote the solution to $Ax = b$. There exists a starting point $x_0 \in \mathbb{R}^n$ such that the sequence (3.19) satisfies for all $k \geq 0$,*

$$\mathbb{E}[\|x_k - x_*\|_A] \geq (1 - \mu)^k \|x_0 - x_*\|_A \,. \tag{3.27}$$

From (3.20) we see that Gauss-Seidel using random coordinates computes a solution x_k satisfying $\mathbb{E}[\|x_k - x_*\|_{A_{1,\beta}}] \leq \varepsilon$ in at most $k = O(\frac{n}{p}\log(1/\varepsilon))$ iterations. On the other hand, Proposition 3.4.3 states that for any fixed partition, there exists an input x_0 such that $k = \Omega(\beta\log(1/\varepsilon))$ iterations are required to reach the same ε error tolerance. Furthermore, the situation does not improve even if use ACDM from [48]. Proposition 3.4.6, which we describe later, implies that for any fixed partition there exists an input x_0 such that $k = \Omega\left(\sqrt{\frac{n}{p}\beta}\log(1/\varepsilon)\right)$ iterations are required for ACDM to reach ε error. Hence as $\beta \longrightarrow \infty$, the gap between random coordinate and fixed partitioning can be made arbitrarily large. These findings are numerically verified in Section 3.4.5.1.

3.4.3.2 A Lyapunov analysis of accelerated Gauss-Seidel and Kaczmarz

Motivated by our findings, our goal is to understand the behavior of accelerated Gauss-Seidel under random coordinate sampling. In order to do this, we establish a general framework from which the behavior of accelerated Gauss-Seidel with random coordinate sampling follows immediately, along with rates for accelerated randomized Kaczmarz [33] and the accelerated coordinate descent methods of [48] and [1].

For conciseness, we describe a simpler version of our framework which is still able to capture both the Gauss-Seidel and Kaczmarz results, deferring the general version to the full version of the paper. Our general result requires a bit more notation, but follows the same line of reasoning.

Let H be a random $n \times n$ positive semi-definite matrix. Put $G = \mathbb{E}[H]$, and suppose that G exists and is positive definite. Furthermore, let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be strongly convex and smooth, and let μ denote the strong convexity constant of f w.r.t. the $\|\cdot\|_{G^{-1}}$ norm.

Consider the following sequence $\{(x_k, y_k, z_k)\}_{k\geq 0}$ defined by the recurrence

$$x_{k+1} = \frac{1}{1+\tau}y_k + \frac{\tau}{1+\tau}z_k\,, \tag{3.28a}$$

$$y_{k+1} = x_{k+1} - H_k\nabla f(x_{k+1})\,, \tag{3.28b}$$

$$z_{k+1} = z_k + \tau(x_{k+1} - z_k) - \frac{\tau}{\mu}H_k\nabla f(x_{k+1})\,, \tag{3.28c}$$

where $H_0, H_1, \ldots$ are independent realizations of H and τ is a parameter to be chosen. Following [77], we construct a candidate Lyapunov function E_k for the sequence (3.28) defined as

$$E_k = f(y_k) - f_* + \frac{\mu}{2}\|z_k - x_*\|^2_{G^{-1}}\,. \tag{3.29}$$

The following theorem demonstrates that E_k is indeed a Lyapunov function for (x_k, y_k, z_k).

Theorem 3.4.4. *Let f, G, H be as defined above. Suppose further that f has 1-Lipschitz gradients w.r.t. the $\|\cdot\|_{G^{-1}}$ norm, and for every fixed $x \in \mathbb{R}^n$,*

$$f(\Phi(x; H)) \leq f(x) - \frac{1}{2}\|\nabla f(x)\|^2_H\,, \tag{3.30}$$

holds for a.e. H, where $\Phi(x;H) = x - H\nabla f(x)$. Set τ in (3.28) as $\tau = \sqrt{\mu/\nu}$, with

$$\nu = \lambda_{\max}\left(\mathbb{E}\left[(G^{-1/2}HG^{-1/2})^2\right]\right) .$$

Then for every $k \geq 0$, we have

$$\mathbb{E}[E_k] \leq (1-\tau)^k E_0 .$$

We now proceed to specialize Theorem 3.4.4 to both the Gauss-Seidel and Kaczmarz settings.

Accelerated Gauss-Seidel Let $S \in \mathbb{R}^{n\times p}$ denote a random sketching matrix. As suggested in Section 3.4.2, we set $f(x) = \frac{1}{2}x^\mathsf{T}Ax - x^\mathsf{T}b$ and put $H = S(S^\mathsf{T}AS)^\dagger S^\mathsf{T}$. Note that $G = \mathbb{E}[S(S^\mathsf{T}AS)^\dagger S^\mathsf{T}]$ is positive definite iff $\lambda_{\min}(\mathbb{E}[P_{A^{1/2}S}]) > 0$, and is hence satisfied for both fixed partition and random coordinate sampling (c.f. Section 3.4.2). Next, the fact that f is 1-Lipschitz w.r.t. the $\|\cdot\|_{G^{-1}}$ norm and the condition (3.30) are standard calculations. All the hypotheses of Theorem 3.4.4 are thus satisfied, and the conclusion is Theorem 3.4.5, which characterizes the rate of convergence for accelerated Gauss-Seidel (Algorithm 1).

Algorithm 1 Accelerated randomized block Gauss-Seidel.

Require: $A \in \mathbb{R}^{n\times n}$, $A \succ 0$, $b \in \mathbb{R}^n$, sketching matrices $\{S_k\}_{k=0}^{T-1} \subseteq \mathbb{R}^{n\times p}$, $x_0 \in \mathbb{R}^n$, $\mu \in (0,1)$, $\nu \geq 1$.

Set $\tau = \sqrt{\mu/\nu}$.
Set $y_0 = z_0 = x_0$.
for $k = 0, ..., T-1$ **do**
  $x_{k+1} = \frac{1}{1+\tau}y_k + \frac{\tau}{1+\tau}z_k$.
  $H_k = S_k(S_k^\mathsf{T}AS_k)^\dagger S_k^\mathsf{T}$.
  $y_{k+1} = x_{k+1} - H_k(Ax_{k+1} - b)$.
  $z_{k+1} = z_k + \tau(x_{k+1} - z_k) - \frac{\tau}{\mu}H_k(Ax_{k+1} - b)$.
end for
Return y_T.

Theorem 3.4.5. *Let A be an $n \times n$ positive definite matrix and $b \in \mathbb{R}^n$. Let $x_* \in \mathbb{R}^n$ denote the unique vector satisfying $Ax_* = b$. Suppose each S_k, $k = 0, 1, 2, ...$ is an independent copy of a random matrix $S \in \mathbb{R}^{n\times p}$. Put $\mu = \lambda_{\min}(\mathbb{E}[P_{A^{1/2}S}])$, and suppose the distribution of S satisfies $\mu > 0$. Invoke Algorithm 1 with μ and ν, where*

$$\nu = \lambda_{\max}\left(\mathbb{E}\left[(G^{-1/2}HG^{-1/2})^2\right]\right) , \tag{3.31}$$

with $H = S(S^\mathsf{T}AS)^\dagger S^\mathsf{T}$ and $G = \mathbb{E}[H]$. Then with $\tau = \sqrt{\mu/\nu}$, for all $k \geq 0$,

$$\mathbb{E}[\|y_k - x_*\|_A] \leq \sqrt{2}(1-\tau)^{k/2}\|x_0 - x_*\|_A . \tag{3.32}$$

Note that in the setting of Theorem 3.4.5, by the definition of ν and μ, it is always the case that $\nu \leq 1/\mu$. Therefore, the iteration complexity of acceleration is at least as good as the iteration complexity without acceleration.

We conclude our discussion of Gauss-Seidel by describing the analogue of Proposition 3.4.3 for Algorithm 1, which shows that our analysis in Theorem 3.4.5 is tight order-wise. The following proposition applies to ACDM as well; we show in the full version of the paper how ACDM can be viewed as a special case of Algorithm 1.

Proposition 3.4.6. *Under the setting of Theorem 3.4.5, there exists starting positions $y_0, z_0 \in \mathbb{R}^n$ such that the iterates $\{(y_k, z_k)\}_{k\geq 0}$ produced by Algorithm 1 satisfy*

$$\mathbb{E}[\|y_k - x_*\|_A + \|z_k - x_*\|_A] \geq (1-\tau)^k \|y_0 - x_*\|_A \,.$$

Accelerated Kaczmarz The argument for Theorem 3.4.5 can be slightly modified to yield a result for randomized accelerated Kaczmarz in the sketching framework, for the case of a consistent overdetermined linear system.

Specifically, suppose we are given an $m \times n$ matrix A which has full column rank, and $b \in \mathcal{R}(A)$. Our goal is to recover the unique x_* satisfying $Ax_* = b$. To do this, we apply a similar line of reasoning as [29]. We set $f(x) = \frac{1}{2}\|x - x_*\|_2^2$ and $H = P_{A^\mathsf{T} S}$, where S again is our random sketching matrix. At first, it appears our choice of f is problematic since we do not have access to f and ∇f, but a quick calculation shows that $H\nabla f(x) = (S^\mathsf{T} A)^\dagger S^\mathsf{T}(Ax - b)$. Hence, with $r_k = Ax_k - b$, the sequence (3.28) simplifies to

$$x_{k+1} = \frac{1}{1+\tau} y_k + \frac{\tau}{1+\tau} z_k \,, \tag{3.33a}$$

$$y_{k+1} = x_{k+1} - (S_k^\mathsf{T} A)^\dagger S_k^\mathsf{T} r_{k+1} \,, \tag{3.33b}$$

$$z_{k+1} = z_k + \tau(x_{k+1} - z_k) - \frac{\tau}{\mu}(S_k^\mathsf{T} A)^\dagger S_k^\mathsf{T} r_{k+1} \,. \tag{3.33c}$$

The remainder of the argument proceeds nearly identically, and leads to the following theorem.

Theorem 3.4.7. *Let A be an $m \times n$ matrix with full column rank, and $b = Ax_*$. Suppose each S_k, $k = 0, 1, 2, \ldots$ is an independent copy of a random sketching matrix $S \in \mathbb{R}^{m\times p}$. Put $H = P_{A^\mathsf{T} S}$ and $G = \mathbb{E}[H]$. The sequence (3.33) with $\mu = \lambda_{\min}(\mathbb{E}[P_{A^\mathsf{T} S}])$, $\nu = \lambda_{\max}\left(\mathbb{E}\left[(G^{-1/2} H G^{-1/2})^2\right]\right)$, and $\tau = \sqrt{\mu/\nu}$ satisfies for all $k \geq 0$,*

$$\mathbb{E}[\|y_k - x_*\|_2] \leq \sqrt{2}(1-\tau)^{k/2} \|x_0 - x_*\|_2 \,. \tag{3.34}$$

Specialized to the setting of [33] where each row of A has unit norm and is sampled uniformly at every iteration, it can be shown (Section C.5.1) that $\nu \leq m$ and $\mu = \frac{1}{m}\lambda_{\min}(A^\mathsf{T} A)$. Hence, the above theorem states that the iteration complexity to reach ε error is $O\left(\frac{m}{\sqrt{\lambda_{\min}(A^\mathsf{T} A)}} \log(1/\varepsilon)\right)$, which matches Theorem 5.1 of [33] order-wise. However, Theorem 3.4.7 applies in general for any sketching matrix.

3.4.3.3 Specializing accelerated Gauss-Seidel to random coordinate sampling

We now instantiate Theorem 3.4.5 to random coordinate sampling. The μ quantity which appears in Theorem 3.4.5 is identical to the quantity appearing in the rate (3.20) of non-accelerated Gauss-Seidel. That is, the iteration complexity to reach tolerance ε is

$$O\left(\sqrt{\nu\mu_{\text{rand}}^{-1}}\log(1/\varepsilon)\right),$$

and the only new term here is ν. In order to provide a more intuitive interpretation of the ν quantity, we present an upper bound on ν in terms of an effective block condition number defined as follows. Given an index set $J \subseteq 2^{[n]}$, define the effective block condition number of a matrix A as $\kappa_{\text{eff},J}(A) = \frac{\max_{i\in J} A_{ii}}{\lambda_{\min}(A_J)}$. Note that $\kappa_{\text{eff},J}(A) \leq \kappa(A_J)$ always. The following lemma gives upper and lower bounds on the ν quantity.

Lemma 3.4.8. *Let A be an $n \times n$ positive definite matrix and let p satisfy $1 < p < n$. We have that*

$$\frac{n}{p} \leq \nu \leq \frac{n}{p}\left(\frac{p-1}{n-1} + \left(1 - \frac{p-1}{n-1}\right)\kappa_{\text{eff},p}(A)\right),$$

where $\kappa_{\text{eff},p}(A) = \max_{J\subseteq 2^{[n]}:|J|=p} \kappa_{\text{eff},J}(A)$, ν is defined in (3.31), and the distribution of S corresponds to uniformly selecting p coordinates without replacement.

Lemma 3.4.8 states that if the $p\times p$ sub-blocks of A are well-conditioned as defined by the effective block condition number $\kappa_{\text{eff},J}(A)$, then the speed-up of accelerated Gauss-Seidel with random coordinate selection over its non-accelerate counterpart parallels the case of fixed partitioning sampling (i.e. the rate described in (3.24) versus the rate in (3.20)). This is a reasonable condition, since very ill-conditioned sub-blocks will lead to numerical instabilities in solving the sub-problems when implementing Gauss-Seidel. On the other hand, we note that Lemma 3.4.8 provides merely a sufficient condition for speed-ups from acceleration, and is conservative. Our numerically experiments in Section 3.4.5.6 suggest that in many cases the ν parameter behaves closer to the lower bound n/p than Lemma 3.4.8 suggests. We leave a more thorough theoretical analysis of this parameter to future work.

We can now combine Theorem 3.4.5 with (3.23) to derive the following upper bound on the iteration complexity of accelerated Gauss-Seidel with random coordinates as

$$N_{\text{rand,acc}} \leq O\left(\frac{n}{p}\sqrt{\frac{\max_{1\leq i\leq n} A_{ii}}{\lambda_{\min}(A)}\kappa_{\text{eff},p}(A)}\log(1/\varepsilon)\right).$$

Illustrative example. We conclude our results by illustrating our bounds on a simple example. Consider the sub-family $\{A_\delta\}_{\delta>0} \subseteq \mathscr{A}$, with

$$A_\delta = A_{n+\delta,-n}\,, \quad \delta > 0\,. \tag{3.35}$$

A simple calculation yields that $\kappa_{\text{eff},p}(A_\delta) = \frac{n-1+\delta}{n-p+\delta}$, and hence Lemma 3.4.8 states that $\nu(A_\delta) \leq \frac{n}{p}\left(1 + \frac{p-1}{n-1}\right)$. Furthermore, by a similar calculation to Proposition 3.4.2, $\mu_{\text{rand}} = \frac{p\delta}{n(n-p+\delta)}$. Assuming for simplicity that $p = o(n)$ and $\delta \in (0, 1)$, Theorem 3.4.5 states that at most $O(\frac{n^{3/2}}{p\sqrt{\delta}} \log(1/\varepsilon))$ iterations are sufficient for an ε-accurate solution. On the other hand, without acceleration (3.20) states that $O(\frac{n^2}{p\delta} \log(1/\varepsilon))$ iterations are sufficient and Proposition 3.4.3 shows there exists a starting position for which it is necessary. Hence, as either n grows large or δ tends to zero, the benefits of acceleration become more pronounced.

3.4.4 Related Work

We split the related work into two broad categories of interest: (a) work related to coordinate descent (CD) methods on convex functions and (b) randomized solvers designed for solving consistent linear systems.

When A is positive definite, Gauss-Seidel can be interpreted as an instance of coordinate descent on a strongly convex quadratic function. We therefore review related work on both non-accelerated and accelerated coordinate descent, focusing on the randomized setting instead of the more classical cyclic order or Gauss-Southwell rule for selecting the next coordinate. See [71] for a discussion on non-random selection rules, [49] for a comparison of random selection versus Gauss-Southwell, and [50] for efficient implementations of Gauss-Southwell.

Nesterov's original paper in [42] first considered randomized CD on convex functions, assuming a partitioning of coordinates fixed ahead of time. The analysis included both non-accelerated and accelerated variants for convex functions. This work sparked a resurgence of interest in CD methods for large problems. Most relevant to our paper are extensions to the block setting [63], handling arbitrary sampling distributions [55, 56, 18], and second order updates for quadratic functions [57].

For accelerated CD, Lee and Sidford [29] generalize the analysis of Nesterov [42]. While the analysis of [29] was limited to selecting a single coordinate at a time, several follow on works [55, 32, 35, 17] generalize to block and non-smooth settings. More recently, both Allen-Zhu et al. [1] and Nesterov and Stich [48] independently improve the results of [29] by using a different non-uniform sampling distribution. One of the most notable aspects of the analysis in [1] is a departure from the (probabilistic) estimate sequence framework of Nesterov. Instead, the authors construct a valid Lyapunov function for coordinate descent, although they do not explicitly mention this. In our work, we make this Lyapunov point of view explicit. The constants in our acceleration updates arise from a particular discretization and Lyapunov function outlined from Wilson et al. [77]. Using this framework makes our proof particularly transparent, and allows us to recover results for strongly convex functions from [1] and [48] as a special case.

From the numerical analysis side both the Gauss-Seidel and Kaczmarz algorithm are classical methods. Strohmer and Vershynin [69] were the first to prove a linear rate of convergence for randomized Kaczmarz, and Leventhal and Lewis [31] provide a similar kind of

analysis for randomized Gauss-Seidel. Both of these were in the single constraint/coordinate setting. The block setting was later analyzed by Needell and Tropp [38]. More recently, Gower and Richtárik [20] provide a unified analysis for both randomized block Gauss-Seidel and Kaczmarz in the sketching framework. We adopt this framework in this paper. Finally, Liu and Wright [33] provide an accelerated analysis of randomized Kaczmarz once again in the single constraint setting and we extend this to the block setting.

3.4.5 Experiments

In this section we experimentally validate our theoretical results on how our accelerated algorithms can improve convergence rates. Our experiments use a combination of synthetic matrices and matrices from large scale machine learning tasks.

Setup. We run all our experiments on a 4 socket Intel Xeon CPU E7-8870 machine with 18 cores per socket and 1TB of DRAM. We implement all our algorithms in Python using `numpy`, and use the Intel MKL library with 72 OpenMP threads for numerical operations. We report errors as relative errors, i.e. $\|x_k - x_*\|_A^2 / \|x_*\|_A^2$. Finally, we use the best values of μ and ν found by tuning each experiment.

We implement fixed partitioning by creating random blocks of coordinates at the beginning of the experiment and cache the corresponding matrix blocks to improve performance. For random coordinate sampling, we select a new block of coordinates at each iteration.

For our fixed partition experiments, we restrict our attention to uniform sampling. While Gower and Richtárik [20] propose a non-uniform scheme based on $\mathbf{Tr}(S^\mathsf{T} A S)$, for translation-invariant kernels this reduces to uniform sampling. Furthermore, as the kernel block Lipschitz constants were also roughly the same, other non-uniform schemes [1] also reduce to nearly uniform sampling.

3.4.5.1 Fixed partitioning vs random coordinate sampling

Our first set of experiments numerically verify the separation between fixed partitioning sampling versus random coordinate sampling.

Figure 3.1 shows the progress per iteration on solving $A_{1,\beta} x = b$, with the $A_{1,\beta}$ defined in Section 3.4.3.1. Here we set $n = 5000$, $p = 500$, $\beta = 1000$, and $b \sim N(0, I)$. Figure 3.1 verifies our analytical findings in Section 3.4.3.1, that the fixed partition scheme is substantially worse than uniform sampling on this instance. It also shows that in this case, acceleration provides little benefit in the case of random coordinate sampling. This is because both μ and $1/\nu$ are order-wise p/n, and hence the rate for accelerated and non-accelerated coordinate descent coincide. However we note that this only applies for matrices where μ is as large as it can be (i.e. p/n), that is instances for which Gauss-Seidel is already converging at the optimal rate (see [20], Lemma 4.2).

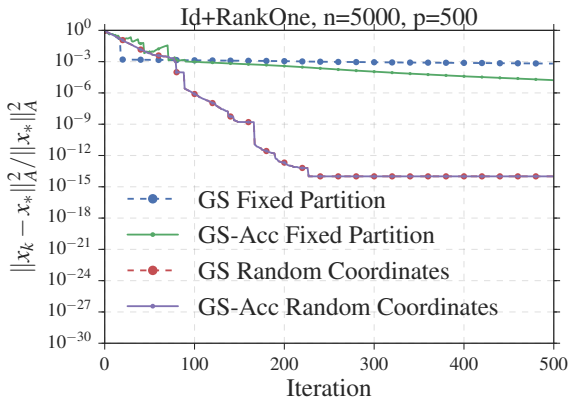

Figure 3.1: Experiments comparing fixed partitions versus random coordinate sampling for the example from Section 3.4.3.1 with $n = 5000$ coordinates, block size $p = 500$.

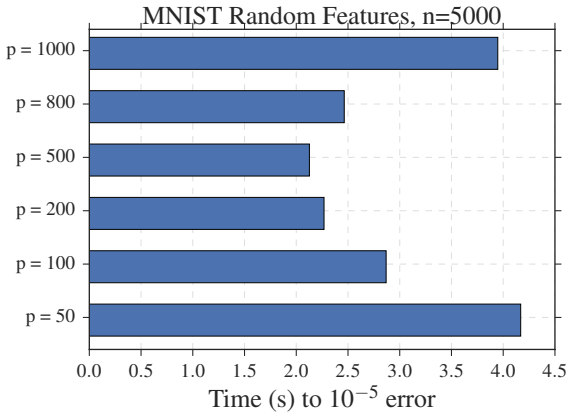

Figure 3.2: The effect of block size on the accelerated Gauss-Seidel method. For the MNIST dataset (pre-processed using random features) we see that block size of $p = 500$ works best.

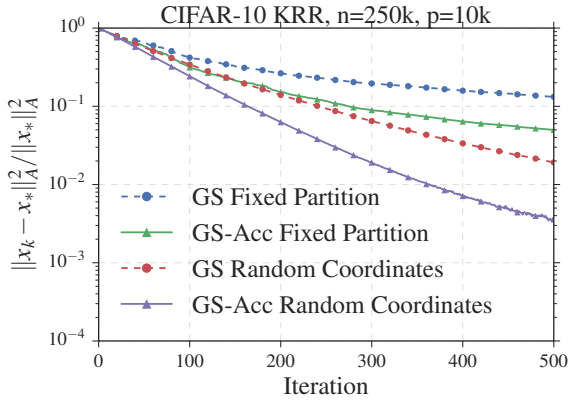

Figure 3.3: Experiments comparing fixed partitions versus uniform random sampling for CIFAR-10 augmented matrix while running kernel ridge regression. The matrix has $n = 250000$ coordinates and we set block size to $p = 10000$.

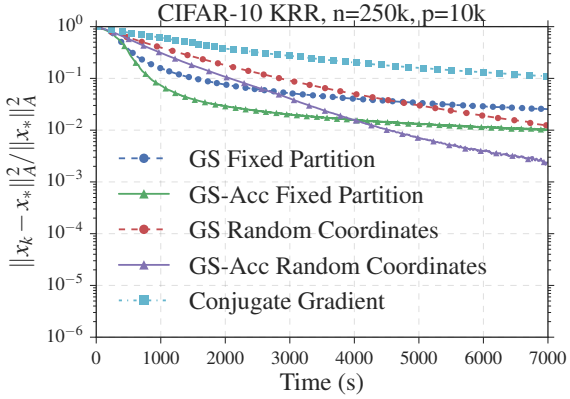

Figure 3.4: Comparing conjugate gradient with accelerated and un-accelerated Gauss-Seidel methods for CIFAR-10 augmented matrix while running kernel ridge regression. The matrix has $n = 250000$ coordinates and we set block size to $p = 10000$.

3.4.5.2 Kernel ridge regression

We next evaluate how fixed partitioning and random coordinate sampling affects the performance of Gauss-Seidel on large scale machine learning tasks. We use the popular image

classification dataset CIFAR-10 and evaluate a kernel ridge regression (KRR) task with a Gaussian kernel. Specifically, given a labeled dataset $\{(x_i, y_i)\}_{i=1}^n$, we solve the linear system $(K + \lambda I)\alpha = Y$ with $K_{ij} = \exp(-\gamma\|x_i - x_j\|_2^2)$, where $\lambda, \gamma > 0$ are tunable parameters (see e.g. [67] for background on KRR). The key property of KRR is that the kernel matrix K is positive semi-definite, and hence Algorithm 1 applies.

For the CIFAR-10 dataset, we augment the dataset[1] to include five reflections, translations per-image and then apply standard pre-processing steps used in image classification [12, 68]. We finally apply a Gaussian kernel on our pre-processed images and the resulting kernel matrix has $n = 250000$ coordinates.

Results from running 500 iterations of random coordinate sampling and fixed partitioning algorithms are shown in Figure 3.3. Comparing convergence across iterations, similar to previous section, we see that un-accelerated Gauss-Seidel with random coordinate sampling is better than accelerated Gauss-Seidel with fixed partitioning. However we also see that using acceleration with random sampling can further improve the convergence rates, especially to achieve errors of 10^{-3} or lower.

We also compare the convergence with respect to running time in Figure 3.4. Fixed partitioning has better performance in practice random access is expensive in multi-core systems. However, we see that this speedup in implementation comes at a substantial cost in terms of convergence rate. For example in the case of CIFAR-10, using fixed partitions leads to an error of 1.2×10^{-2} after around 7000 seconds. In comparison we see that random coordinate sampling achieves a similar error in around 4500 seconds and is thus $1.5\times$ faster. We also note that this speedup increases for lower error tolerances.

3.4.5.3 Comparing Gauss-Seidel to Conjugate-Gradient

We also compared Gauss-Seidel with random coordinate sampling to the classical conjugate-gradient (CG) algorithm. CG is an important baseline to compare with, as it is the de-facto standard iterative algorithm for solving linear systems in the numerical analysis community. While we report the results of CG without preconditioning, we remark that the performance using a standard banded preconditioner was not any better. However, for KRR specifically, there have been recent efforts [5, 65] to develop better preconditioners, and we leave a more thorough comparison for future work. The results of our experiment are shown in Figure 3.4. We note that Gauss-Seidel both with and without acceleration outperform CG. As an example, we note that to reach error 10^{-1} on CIFAR-10, CG takes roughly 7000 seconds, compared to less than 2000 seconds for accelerated Gauss-Seidel, which is a $3.5\times$ improvement.

To understand this performance difference, we recall that our matrices A are fully dense, and hence each iteration of CG takes $O(n^2)$. On the other hand, each iteration of both non-accelerated and accelerated Gauss-Seidel takes $O(np^2 + p^3)$. Hence, as long as $p = O(n^{2/3})$, the time per iteration of Gauss-Seidel is order-wise no worse than CG. In terms of

[1] Similar to https://github.com/akrizhevsky/cuda-convnet2.

iteration complexity, standard results state that CG takes at most $O(\sqrt{\kappa}\log(1/\varepsilon))$ iterations to reach an ε error solution, where κ denotes the condition number of A. On the other hand, Gauss-Seidel takes at most $O(\frac{n}{p}\kappa_{\text{eff}}\log(1/\varepsilon))$, where $\kappa_{\text{eff}} = \frac{\max_{1\le i\le n} A_{ii}}{\lambda_{\min}(A)}$. In the case of any (normalized) kernel matrix associated with a translation-invariant kernel such as the Gaussian kernel, we have $\max_{1\le i\le n} A_{ii} = 1$, and hence generally speaking κ_{eff} is much lower than κ.

3.4.5.4 Kernel ridge regression on smaller datasets

In addition to using the large CIFAR-10 augmented dataset, we also tested our algorithms on the smaller MNIST[2] dataset. To generate a kernel matrix, we applied the Gaussian kernel on the raw MNIST pixels to generate a matrix K with $n = 60000$ rows and columns.

Results from running 500 iterations of random coordinate sampling and fixed partitioning algorithms are shown in Figure 3.5. We plot the convergence rates both across time and across iterations. Comparing convergence across iterations we see that random coordinate sampling is essential to achieve errors of 10^{-4} or lower. In terms of running time, similar to the CIFAR-10 experiment, we see that the benefits in fixed partitioning of accessing coordinates faster comes at a cost in terms of convergence rate, especially to achieve errors of 10^{-4} or lower.

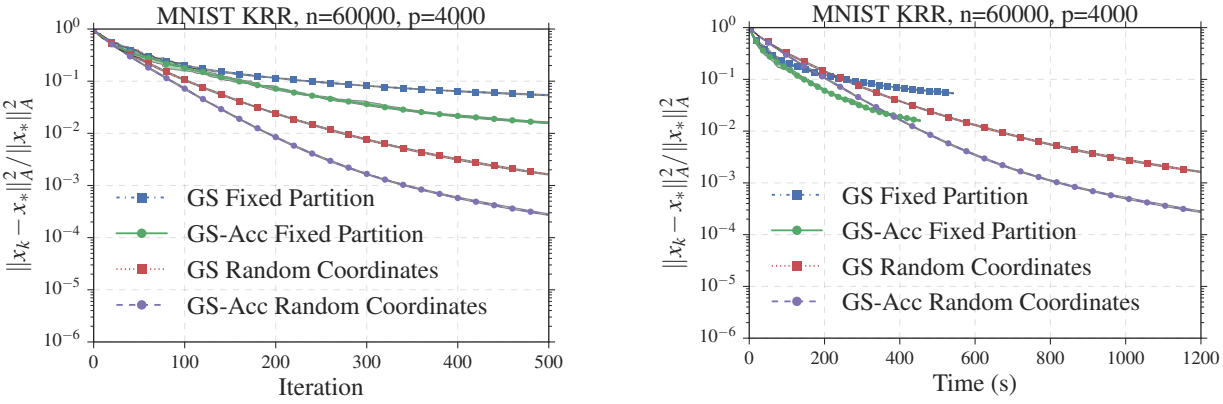

Figure 3.5: Experiments comparing fixed partitions versus uniform random sampling for MNIST while running kernel ridge regression. MNIST has $n = 60000$ coordinates and we set block size to $p = 4000$.

3.4.5.5 Effect of block size

We next analyze the importance of the block size p for the accelerated Gauss-Seidel method. As the values of μ and ν change for each setting of p, we use a smaller MNIST matrix for this experiment. We apply a random feature transformation [60] to generate an $n \times d$ matrix

[2] http://yann.lecun.com/exdb/mnist/

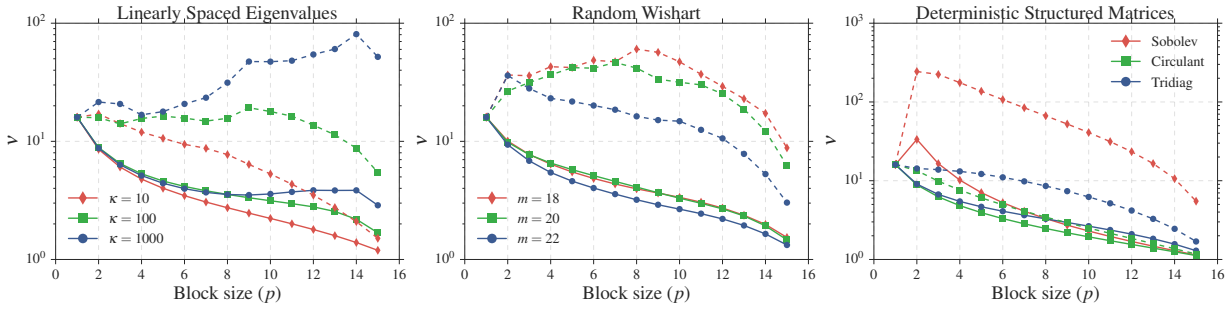

Figure 3.6: Comparison of the computed ν constant (solid lines) and ν bound from Theorem 3.4.5 (dotted lines) on random matrices with linearly spaced eigenvalues and random Wishart matrices.

F with $d = 5000$ features. We then use $A = F^\mathsf{T}F$ and $b = F^\mathsf{T}Y$ as inputs to the algorithm. Figure 3.2 shows the wall clock time to converge to 10^{-5} error as we vary the block size from $p = 50$ to $p = 1000$.

Increasing the block-size improves the amount of progress that is made per iteration but the time taken per iteration increases as $O(p^3)$ (Line 5, Algorithm 1). However, using efficient BLAS-3 primitives usually affords a speedup from systems techniques like cache blocking. We see the effects of this in Figure 3.2 where using $p = 500$ performs better than using $p = 50$. We also see that these benefits reduce for much larger block sizes and thus $p = 1000$ is slower.

3.4.5.6 Computing the μ and ν constants

In our last experiment, we explicitly compute the μ and ν constants from Theorem 3.4.5 for a few 16×16 positive definite matrices constructed as follows.

Linearly spaced eigenvalues. We first draw Q uniformly at random from $n \times n$ orthogonal matrices. We then construct $A_i = Q\Sigma_i Q^\mathsf{T}$ for $i = 1, 2, 3$, where Σ_1 is `diag(linspace(1, 10, 16))`, Σ_2 is `diag(linspace(1, 100, 16))`, and Σ_3 is `diag(linspace(1, 1000, 16))`.

Random Wishart. We first draw B_i with iid $N(0, 1)$ entries, where $B_i \in \mathbb{R}^{m_i \times 16}$ with $m_1 = 18$, $m_2 = 20$, and $m_3 = 22$. We then set $A_i = B_i^\mathsf{T} B_i$.

Sobolev kernel. We form the matrix $A_{ij} = \min(i, j)$ with $1 \le i, j \le n$. This corresponds to the gram matrix for the set of points $x_1, ..., x_n \in \mathbb{R}$ with $x_i = i$ under the Sobolev kernel $\min(x, y)$.

Circulant matrix. We let A be a 16×16 instance of the family of circulant matrices $A_n = F_n \mathrm{diag}(c_n) F_n^*$ where F_n is the $n \times n$ unitary DFT matrix and $c_n = (1, 1/2, ..., 1/(n/2+1), ..., 1/2, 1)$. By construction this yields a real valued circulant matrix which is positive definite.

Tridiagonal matrix. We let A be a tridiagonal matrix with the diagonal value equal to one, and the off diagonal value equal to $(\delta - a)/(2\cos(\pi n/(n+1)))$ for $\delta = 1/10$. The matrix

has a minimum eigenvalue of δ.

Figure 3.6 shows the results of our computation for the linearly spaced eigenvalues ensemble, the random Wishart ensemble and the other deterministic structured matrices. Alongside with the actual ν values, we plot the bound given for each instance by Lemma 3.4.8. From the figures we see that our bound is quite close to the computed value of ν for circulant matrices and for random matrices with linearly spaced eigenvalues with small κ. We plan to extend our analysis to derive a tighter bound in the future.

3.4.6 Conclusion

In this paper, we extended the accelerated block Gauss-Seidel algorithm beyond fixed partition sampling. Our analysis introduced a new data-dependent parameter ν which governs the speed-up of acceleration. Specializing our theory to random coordinate sampling, we derived an upper bound on ν which shows that well conditioned blocks are a sufficient condition to ensure speedup. Experimentally, we showed that random coordinate sampling is readily accelerated beyond what our bound suggests.

The most obvious question remains to derive a sharper bound on the ν constant from Theorem 3.4.5. Another interesting question is whether or not the iteration complexity of random coordinate sampling is always bounded above by the iteration complexity with fixed coordinate sampling.

We also plan to study an implementation of accelerated Gauss-Seidel in a distributed setting [72]. The main challenges here are in determining how to sample coordinates without significant communication overheads, and to efficiently estimate μ and ν. To do this, we wish to explore other sampling schemes such as shuffling the coordinates at the end of every epoch [62].

3.5 Summary

The Lyapunov framework we have discussed in this thesis is an especially nice way of viewing convergence theory in optimization. Many algorithms in optimization are significantly less mysterious when viewed through the lens of dynamical systems. The Lyapunov framework also makes the introduction of new analyses seamless. As a simple example, most of the non-asymptotic results for coordinate methods can be extended to geodesic spaces, for geodesic (smooth/strong)-convex functions, using the Lyapunov framework. There are many other examples as well.

Appendix A

Chapter One

A.1 Examples of Optimization Problems

Example A.1.1 (Planning)**.** *The decision variable is an action, the set $\mathcal{X}$ represents all the actions under consideration, and the objective function $f : \mathcal{X} \to \mathbb{R}$ assigns a cost to each action. The optimization problem is to find an action with minimal cost.*

As a specific example, consider the problem of reasoning about choosing a path to reach a destination. In this instance, $\mathcal{X}$ is the set of easible paths from our starting position to the desired location and the functions $f(x)$ measures the cost to travel on path x. Solving the optimization problem finds the path with minimal cost. Another example is choosing a project from a set of proposals $\mathcal{X}$. In this example, the function maps each proposal to an estimate of its negative profit. Solving the optimization problem finds the project that maximizes the profit, $x^* \in \arg\min_{x \in \mathcal{X}} f(x)$.

Example A.1.2 (Empricial Risk Minimization)**.** *The decision variable x is a function, the set $\mathcal{X}$ imposes assumptions on the function class, and the objective function is the prediction error when evaluated on an observe dataset. The optimization problem is to find the function x with minimal prediction error on the observed data.*

As a specific example, consider the goal of providing images with short descriptions, or labels. In machine learning, the task is posed as finding a function $x^* \in \arg\min_{x \in \mathcal{X}} f(x)$ that minimizes a functional, where each function $x \in \mathcal{X}$ maps images to labels. The functional to be minimized, called the *empirical risk*, measures the error of each function $x \in \mathcal{X}$ when evaluated on some observed data z. Thus, the problem is posed as finding the function which performs best on the observed dataset, with the hope that the classifier will perform well more generally.

Example A.1.3 (Maximum likelihood)**.** *The decision variable x is a vector of parameters of a probabilistic model, the $\mathcal{X}$ is the set of possible parameters, and the objective function f is the negative log-likelihood of the observed data. The optimization problem is to find the parameters x which maximizes the likelihood of the observed data*

As a specific example, consider the inverse problem in scientific measurement. In this example, the negative log-likelihood $f(x) = -\log p(z; x)$ encodes the probability of observing the noisy measurement z when the ground-truth object is x. The maximum likelihood problem is to find the ground-truth object x, from some restricted set $\mathcal{X}$, that maximizes the likelihood of observing z.

Example A.1.4 (Robust/worst-case Planning)**.** *The decision variable x is an action being taken by some adversarial entity, the set $\mathcal{X}$ are the actions the adversary can make, and the objective function f is the gain or loss received due to the adversary's actions. The optimization problem is to find the worst-case loss due to an adversary's actions.*

A.2 Glossary of Definitions

In this section we discuss several basic concepts from calculus and geometry that will be used repeatedly in proofs of convergence. For many readers, most of this material will be familiar and can be referred to only when necessary.

Calculus In this section we take $\mathcal{X} = \mathbb{R}^d$, but many of the definitions we discuss can extended to convex, compact sets $\mathcal{X} \subseteq \mathbb{R}^d$ in a natural way. Let $f : \mathbb{R}^d \to \mathbb{R}$ be continuously differentiable as many times as necessary. Recall that the Taylor series is a representation of a function as an infinite sum of terms, calculated from the values of the function's derivatives at a single point. In particular, for an integer $p \geq 2$, the $(p-1)$-st order Taylor approximation of f centered at $x \in \mathbb{R}^d$ is the $(p-1)$-st degree polynomial

$$f_{p-1}(y; x) = \sum_{i=0}^{p-1} \frac{1}{i!} \nabla^i f(x)(y-x)^i = f(x) + \langle \nabla f(x), y - x \rangle + \cdots + \frac{1}{(p-1)!} \nabla^{p-1} f(x)(y-x)^{p-1}. \tag{A.1}$$

The Taylor series provides a way to approximate any function using a finite number of polynomial terms. We call the difference between the function f at a point y and its first-order Taylor series approximation at a point x the **Bregman divergence** of f,

$$D_f(y, x) = f(y) - f(x) - \langle \nabla f(x), y - x \rangle. \tag{A.2}$$

Note, $D_f(y, x) \approx \langle y - x, \nabla^2 f(x)(y - x) \rangle$. If f is *convex* its Bregman divergence (A.2) is non-negative,

$$D_f(y, x) \geq 0. \tag{A.3}$$

(A.3) is equivalent to the condition that $\forall x, y \in \mathcal{X}$, any intermediate value is at most the average value (*Jensen's inequality*)

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda) f(y) \tag{A.4}$$

To elaborate, when $\mathcal{X} = \mathbb{R}^d$, convexity is the condition that the first-order Taylor approximation, $f(x) + \langle \nabla f(x), y - x \rangle$, is a *global* underestimator of the function. If $\nabla f(x) = 0$ in (A.2), this implies the property $f(x) \leq f(y)$, $\forall x \in \mathbb{R}^d$, and subsequently that x is a global minimizer of the function. Therefore, if f is convex, the local condition $\nabla f(x) = 0$, implies that x is a solution to (1.1). A function f is *strictly convex* if the inequality in (A.4) and (A.3) is strict ($>$ and not $\geq$) $\forall x \neq y$. For strictly convex functions, the local condition, $\nabla f(x) = 0$, implies the global property, x is the *unique* solution to (1.1).

We say f is μ-uniformly convex ($p \geq 2$) if

$$D_f(y, x) \geq \frac{\mu}{p} \|y - x\|^p. \tag{A.5}$$

When $p = 2$, the condition (A.5) is called *strong convexity*,

$$D_f(y, x) \geq \frac{\mu}{2} \|y - x\|^2. \tag{A.6}$$

Let $h : \mathbb{R}^d \to \mathbb{R}$ be a strictly convex, differentiable function. We say f is μ-strongly convex with respect to a strictly convex function h, if for $\forall x, y \in \mathcal{X}$ it satisfies,

$$D_f(y, x) \geq \mu D_h(y, x). \tag{A.7}$$

If $h(x) = \frac{1}{2}\|x\|^2$ then (A.6) and (A.7) are equivalent. Furthermore, if $h(x) = \frac{1}{p}\|x\|^p$, h satisfies (A.5) with $\mu = 2^{-p+2}$. Therefore if f is strongly convex with respect to the Bregman divergence generated by $h(x) = \frac{1}{p}\|x\|^p$ (A.7), it is satisfies (A.5).

Definition A.2.1 (Geodesic)**.** *We can think of a geodesic as a generalization of the straight-line distance to curved metric spaces $\mathcal{X}$. More precisely, we define the length of a path $\gamma : [0, 1] \to \mathcal{X}$ as*

Definition A.2.2 (Geodesic Space)**.** *If for every pair points $x, y \in \mathcal{X}$ there is a geodesic in $\mathcal{X}$ connecting them, $\mathcal{X}$ is called a geodesic space.*

Definition A.2.3 (Geodesic Convexity)**.** *A function is geodesically convex if for any $x, y \in \mathcal{X}$, a geodesic γ such that $\gamma(0) = x$ and $\gamma(1) = y$, and $t \in [0, 1]$, it holds that*

$$f(y) - f(x) \geq \langle \nabla f(x), \log_x(y) \rangle_x, \tag{A.8}$$

where $\nabla f(x)$ is the gradient of f at x. It is geodesically convex along its i-th component if it satisfies

$$D_f^i(y, x) := f(y) - f(x) - \langle \nabla_i f(x), \log_x(y) \rangle_x \leq 0, \tag{A.9}$$

Definition A.2.4 (Geodesically Smooth)**.** *A function is geodesically L-smooth if for any $x, y \in \mathcal{X}$, a geodesic γ such that $\gamma(0) = x$ and $\gamma(1) = y$, and $t \in [0, 1]$, it holds that,*

$$f(y) - f(x) \leq \langle \nabla f(x), \log_x(y) \rangle_x + \frac{L}{2} d^2(x, y) \tag{A.10}$$

where $\nabla f(x)$ is the gradient of f at x. It is geodesically L_i- smooth along its i-th component if it satisfies

$$f(y) - f(x) \leq \langle \nabla_i f(x), \log_x(y) \rangle_x + \frac{L_i}{2} d^2(x, y) \tag{A.11}$$

Smoothness Smoothness is a property which measures the number of continuous derivatives a function has. In particular, we say say that f is *L-smooth of order p*, where p is a positive integer, if f is p-times continuously differentiable and $\nabla^p f$ is L-Lipschitz, which means for all $x, y \in \mathbb{R}^d$,

$$\|\nabla^p f(y) - \nabla^p f(x)\|_* \leq L\|y - x\|. \tag{A.12}$$

Notable cases are when $p = 0$, in which case the function is *Lipschitz*, and when $p = 1$, in which case the function has Lipschitz gradients. If f satisfies (A.12) with $p = 1$, we call the function *smooth*. If f is smooth we can also write,

$$D_f(y, x) \leq \frac{L}{2}\|y - x\|^2. \tag{A.13}$$

A natural generalization of (A.13) is the condition that f be L-smooth with respect to a strictly convex function h,

$$D_f(y, x) \leq L D_h(y, x). \tag{A.14}$$

A natural generalization of (A.12), is that f have *(ν, L)-Hölder continuous gradients of order p*,

$$\|\nabla^p f(y) - \nabla^p f(x)\|_* \leq L\|y - x\|^\nu. \tag{A.15}$$

where $\nu \in [0, 1]$. If f has *(ν, L)-Hölder continuous gradients of order $p = 2$, we can write*

$$D_f(x, y) \leq \frac{L}{1 + \nu}\|x - y\|^{1+\nu} \tag{A.16}$$

For convex functions on a vector space, the **subgradient** *generalizes the notion of a derivative to functions which are not differentiable. The subdifferential of f at a point x is the set of subgradients,*

$$\partial f(x) := \{g(x) \in \mathbb{R}^d : f(y) \geq f(x) + \langle g(x), y - x \rangle \quad \forall y \in \mathbb{R}^d\}. \tag{A.17}$$

We also write $\|\partial f(x)\|_* := \sup_{g(x) \in \partial f(x)} \|g(x)\|_*$. For any element of the subdifferential $g(x) \in \partial f(x)$, we define the short-hand,

$$D_f^g(y, x) := f(y) - f(x) - \langle g(x), y - x \rangle. \tag{A.18}$$

If f is convex, then (A.18) *is non-negative $D_f^g(y,x) \geq 0$, $\forall g \in \partial f(x)$. Given convex functions are locally Lipschitz, $\partial f(x)$ is compact for any x. Therefore, there always exists a directional subgradient for a convex function.*

We use an analogous notation to (A.18) *to define the Bregman divergence defined for the gradient along the i-th coordinate of $x \in \mathcal{X}$:*

$$D_f^i(y,x) := f(y) - f(x) - \langle \nabla_i f(x), y - x \rangle. \tag{A.19}$$

We say that a function f is convex along its ith-coordinate if,

$$D_f^i(y,x) \geq 0. \tag{A.20}$$

We say a function f is L_i-smooth along its i-th coordinate if,

$$D_f^i(y,x) \leq \frac{L_i}{2}\|y - x\|^2. \tag{A.21}$$

Notation *We denote a discrete-time sequence in lower case, e.g., x_k with $k \geq 0$ an integer. We denote a continuous-time curve in upper case, e.g., X_t with $t \in \mathbb{R}$. An over-dot means derivative with respect to time, i.e., $\dot{X}_t = \frac{d}{dt}X_t$.*

Riemannian Manifolds *For many applications, it is important to be able to optimize over spaces that are more general than $\mathbb{R}^n$. One family of spaces we consider in this thesis are spaces that locally look like $\mathbb{R}^n$, but not globally. A typical example is the sphere, which only locally looks flat. In order to define the notion of continuity of the functions on $\mathcal{X}$, it is important that $\mathcal{X}$ be topological space; this way a notion of "nearby" can be well-defined using open sets. We also require $\mathcal{X}$ have enough structure so that it is easy to tell when $f : \mathcal{X} \to \mathbb{R}$ is smooth. Technically speaking, we assume there exists smooth charts covering $\mathcal{X}$ (i.e. an* atlas*) in addition to the collection of open sets. Such spaces are called* smooth manifolds*.*

The last requirement involves the existence of what is called a vector field, which is a space of vectors tangent to each $x \in \mathcal{X}$; this requirement provides us with the notion of a directional derivative for any $f : \mathcal{X} \to \mathbb{R}$, which will be a requirement for the existence of various families of dynamical systems that find a solution to (1.1)*. We call the subspace of vectors tangent to x the* tangent space*, $\mathsf{T}_x\mathcal{X}$. Finally, we require the existence of a symmetric, positive definite, bilinear form, $\mathrm{g}_x : \mathsf{T}_x\mathcal{X} \times \mathsf{T}_x\mathcal{X} \to \mathbb{R}$, which measures the distance between any two vectors in $\mathsf{T}_x\mathcal{X}$, and which varies smoothly as a function of x. Together, $(\mathcal{X}, \mathrm{g}_{\mathcal{X}})$ is called a* smooth Riemannian manifold*.*

Finally, we mention the cotangent space *$\mathsf{T}_x^*\mathcal{X}$, which is the dual vector space to $\mathsf{T}_x\mathcal{X}$ (i.e. the space of linear functionals on $\mathsf{T}_x\mathcal{X}$). The gradient, for example, is an element of the cotangent space, which defines the directional derivative of f:*

$$\langle \nabla f(x), v \rangle \equiv f'(x,v) := \lim_{\delta \to 0} \frac{f(x + \delta v) - f(x)}{\delta} \tag{A.22}$$

The metric g_x *can be viewed as a mapping between the dual spaces* $\mathsf{T}_x\mathcal{X}$ *and* $\mathsf{T}_x\mathcal{X}^*$,

$$\begin{aligned}\mathrm{g}_x : \mathsf{T}_x\mathcal{X} &\to \mathsf{T}_x\mathcal{X}^* \\ v &\mapsto \mathrm{g}_x(v, \cdot).\end{aligned}$$

Given g_x *is positive definite* $\forall x \in \mathcal{X}$ *and* $dim(\mathsf{T}_x\mathcal{X}) = dim(\mathsf{T}_x\mathcal{X}^*)$, *it has a well-defined inverse* $\mathrm{g}^{-1} : \mathsf{T}_x\mathcal{X}^* \to \mathsf{T}_x\mathcal{X}$. *The inverse metric will be useful for defining vector fields in* $\mathsf{T}_x\mathcal{X}$ *using elements of the cotangent space* $\mathsf{T}_x\mathcal{X}^*$.

We can extend the notion of directional derivatives to non-differentiable functions as well. In particular, the directional subgradient of f, *for* $\forall x, v \in \mathbb{R}^d$ *is a Borel measurable function* $\partial f(x; v)$ *given by,*

$$\langle \partial f(x), v\rangle \equiv \partial f(x; v) := \lim_{\delta \to 0} \frac{f(x + \delta v) - f(x)}{\delta} = \sup_{g \in \partial f(x)} \langle g, v\rangle \tag{A.23}$$

If $f : \mathcal{X} \to \mathbb{R}$ *and* $\mathcal{X} \subset \mathbb{R}^n$ *is convex, then the subgradient is guaranteed to exist,* $\partial f(x) \neq \emptyset$, *for every* $x \in int(\mathcal{X})$ *[64].*

Duality *As suggested by the previous paragraph, the dual correspondence between a function* f *and its* convex dual *(or conjugate) function* f^*, *given by the* Legendre-Fenchel *transform, plays an important role in mathematics and optimization. Formally, if* $\mathcal{X}$ *is a vector space, its dual vector space* $\mathcal{X}^*$ *is the space of linear functionals, which itself forms a vector space under point-wise addition and scalar multiplication.*

Let $f : \mathcal{X} \to \mathbb{R}$. *The Legendre-Fenchel transform of* f *is the function* $f^* : \mathcal{X} \to \mathbb{R}$ *given by,*

$$f^*(g) = \sup_{x \in \mathcal{X}} \{\langle g, x\rangle - f(x)\}. \tag{A.24}$$

As f^* *is the supremum of linear functions, it is convex and* $(f^*)^* = f$. *If* $g \in \mathcal{X}^*$ *is of the form* $g = \nabla f(x)$ *for some* $x \in \mathcal{X}$, *then the supremum in* (A.24) *is achieved, and we obtain the identity,* $f^*(\nabla f(x)) = \langle \nabla f(x), x\rangle - f(x)$. *By differentiating this identity, we are able to conclude that* $\nabla f : \mathcal{X}^* \to \mathcal{X}$ *is the* inverse *of* $\nabla f : \mathcal{X} \to \mathcal{X}^*$, *i.e. that* $\nabla f^*(\nabla f(x)) = x$. *As a specific example of duality, consider the standard inner product* $\langle \cdot, \cdot\rangle$ *on* $\mathbb{R}^d$ *which defines a norm* $\|\cdot\|$ *on* $\mathbb{R}^d$. *Its dual* $\|\cdot\|_*$ *is defined as,* $\|g\|_* = \sup_{x \in \mathbb{R}^d}\{\langle x, g\rangle : \|x\| \leq 1\}$. *More generally, Young's inequality,*

$$\langle g, x\rangle + \frac{1}{p}\|x\|^p \geq -\frac{p-1}{p}\|g\|_*^{\frac{p}{p-1}}, \tag{A.25}$$

demonstrates another special case of the duality relation $\langle g, x\rangle \leq f^*(g) + f(x)$, *where* $f(x) = x^p/p$, $f^*(g) = g^q/q$ *and* $1/p + 1/q = 1$. *A final dual relationship that will be used in several proofs is Cauchy-Schwartz's inequality on a Hilbert space* $\mathcal{X}$,

$$|\langle x, y\rangle| \leq \|x\|_2\|y\|_2, \forall x, y \in \mathcal{X} \tag{A.26}$$

Properties of Bregman Divergences *The Bregman three-point identity,*

$$\langle \nabla h(z) - \nabla h(x), x^* - z \rangle + D_h(x^*, z) = D_h(x^*, x) - D_h(z, x), \tag{A.27}$$

which holds for all $x^*, z, x \in \mathcal{X}$, will be used many times througout the text. We also need to make use of the Bregman projection onto a set $\mathcal{X}$. Formally, let $\mathcal{X} \subseteq \mathbb{R}^n$ and $X_t = \arg\min_{x \in \mathcal{X}} D_h(x, Y_t)$ be the Bregman projection of Y_t onto $\mathcal{X}$. For all $x \in \mathcal{X}$, it follows

$$D_h(x, X_t) \leq D_h(x, Y_t) - D_h(X_t, Y_t) \tag{A.28}$$

Furthermore, the Bregman projection is unique.

Another property of Bregman divergence that will make use of is the following dual relationship

$$D_{h^*}(x, y) = D_h(\nabla h^*(y), \nabla h^*(x)) \tag{A.29}$$

for all $x, y \in \mathcal{X}$, where h^* is the fenchel conjugate (A.24) to h.

Appendix B

Chapter Two

B.1 Gradient Descent

B.1.1 Polyak-Löjasiewicz Condition

If the objective function satisfies the Polyak-Löjasiewicz (PL) condition with parameter μ, we can conclude the following for the optimality gap $\tilde{\mathcal{E}}_t = f(X_t) - f(x^*)$,

$$\frac{d}{dt}\tilde{\mathcal{E}}_t \overset{(2.6)}{=} -\|\nabla f(X_t)\|^2 \overset{(2.1)}{\leq} -2\mu(f(X_t) - f(x^*)) \leq -2\mu\tilde{\mathcal{E}}_t. \tag{B.1}$$

Therefore

$$\mathcal{E}_t = e^{2\mu t}(f(X_t) - f(x^*)), \tag{B.2}$$

is a Lyapunov function for any function which satisfies the PL-condition with parameter μ. One can check, $\frac{d}{dt}\mathcal{E}_t = 2\mu e^{2\mu t}\tilde{\mathcal{E}}_t + e^{2\mu t}\frac{d}{dt}\tilde{\mathcal{E}}_t = e^{2\mu t}(2\mu\tilde{\mathcal{E}}_t + \frac{d}{dt}\tilde{\mathcal{E}}_t \overset{(B.1)}{\leq} 0$. Integrating $\int_0^t \frac{d}{ds}\mathcal{E}_s ds = \mathcal{E}_t - \mathcal{E}_0 \leq 0$ allows us to conclude a $O(e^{-\mu t})$ convergence rate,

$$f(X_t) - f(x^*) \leq e^{-2\mu t}\mathcal{E}_0.$$

We can obtain similar statement for GD and PM.

Gradient Descent *For GD, as long as the function is L-smooth, where $1/\delta \leq L$, and the PL inequality holds, we can conclude the following for the optimality gap, $\tilde{E}_k = f(x_k) - f(x^*)$. We check,*

$$\frac{\tilde{E}_{k+1} - \tilde{E}_k}{\delta} \overset{(2.8)}{\leq} -\frac{2 - \delta L}{2}\|\nabla f(x_k)\|^2 \overset{(2.1)}{\leq} -\mu(2 - \delta L)(f(x_k) - f(x^*)) \leq -\mu(2 - \delta L)\tilde{E}_k$$

Taking $1/\delta = L$ and denoting the inverse coniditon number, $\kappa^{-1} = \mu/L = \mu\delta$, we obtain the bound $\frac{\tilde{E}_{k+1} - \tilde{E}_k}{\delta} \leq -\mu\tilde{E}_k$ from which we can conclude

$$E_k = (1 - (\mu\delta))^{-k}(f(x_k) - f(x^*)),$$

is a Lyapunov function.[1] *Summing allows us to conclude an* $O(e^{-\mu\delta k}) \approx O((e^{-\mu\delta})^k) \approx O((1-\mu\delta)^k)$ *convergence rate,*

$$f(x_k) - f(x^*) \leq (1-\kappa^{-1})^k E_0.$$

Proximal Method *For the PM, similar arguments can be made assuming the Polyak-Löjasiewicz condition by using the discrete optimality gap* $\tilde{E}_k = f(x_k) - f(x^*)$,

$$\frac{\tilde{E}_{k+1} - \tilde{E}_k}{\delta} \overset{(2.9)}{\leq} -\frac{1}{2}\|\nabla f(x_{k+1})\|^2 \overset{(2.1)}{\leq} -\mu(f(x_{k+1}) - f(x^*)) = -\mu\tilde{E}_{k+1}.$$

Thus, the recurrence, $\tilde{E}_{k+1} - \tilde{E}_k \leq -\kappa^{-1}\tilde{E}_{k+1}$ *shows that*

$$E_k = (1+\kappa^{-1})^k (f(x_k) - f(x^*)),$$

is a Lyapunov function, where $\kappa^{-1} = \mu\delta$.[2] *By summing, we can conclude a* $O(e^{-\mu\delta k}) \approx O((e^{\mu\delta})^{-k}) \approx O((1+\mu\delta)^{-k})$ *convergence rate*

$$f(x_k) - f(x^*) \leq (1+\kappa^{-1})^{-k} E_0.$$

In moving from continuous to discrete-time, we lose a factor of two on the bound.

B.1.2 Strongly Convex Functions

When the objective function is μ*-strongly convex* (A.6)*, it satisfies the PL inequality; thus, the analysis from the previous section applies. Another Lyapunov function is the scaled distance function on Euclidean space,*

$$\mathcal{E}_t = e^{\mu t}\frac{1}{2}\|x^* - X_t\|^2. \tag{B.3}$$

To see this, we define $\tilde{\mathcal{E}}_t = \frac{1}{2}\|x^* - X_t\|^2$*, and note*

$$\begin{aligned}\frac{d}{dt}\tilde{\mathcal{E}}_t = -\langle \dot{X}_t, x^* - X_t\rangle &\overset{(2.2)}{=} \langle \nabla f(X_t), x^* - X_t\rangle \\ &\overset{(A.6)}{\leq} -\mu\frac{1}{2}\|x^* - X_t\|^2 = -\mu\tilde{\mathcal{E}}_t;\end{aligned}$$

therefore the Lyapunov property is easy to check: $\frac{d}{dt}\mathcal{E}_t = e^{\mu t}(\frac{d}{dt}\tilde{\mathcal{E}}_t + \mu\tilde{\mathcal{E}}_t) \leq 0$*. The Lyapunov property,* $\mathcal{E}_t \leq \mathcal{E}_0$*, for* (B.3) *allows us to conclude*

$$\frac{1}{2}\|x^* - X_t\|^2 \leq e^{-\mu t}\mathcal{E}_0.$$

We can obtain a similar statement for GD and GF.

[1] One can check, $\frac{E_{k+1}-E_k}{\delta} = (1-\mu\delta)^{-(k+1)}\tilde{E}_{k+1}/\delta - (1-\mu\delta)^{-k}\tilde{E}_k/\delta = (1-\mu\delta)^{-(k+1)}(\mu\tilde{E}_k + \frac{\tilde{E}_{k+1}-\tilde{E}_k}{\delta}) \leq 0$

[2] One can check, $\frac{E_{k+1}-E_k}{\delta} = (1+\mu\delta)^{(k+1)}\tilde{E}_{k+1} - (1+\mu\delta)^k\tilde{E}_k/\delta = (1+\mu\delta)^k(\mu\tilde{E}_{k+1} + \frac{\tilde{E}_{k+1}-\tilde{E}_k}{\delta}) \leq 0$

Gradient Descent *For GD, a similarly scaled distance function,*

$$E_k = (1-\mu\delta)^{-k}\frac{1}{2}\|x^* - x_k\|^2,$$

is a Lyapunov function when f is $(1/\delta)$-smooth and μ strongly convex. We similarly use $\tilde{E}_k = \frac{1}{2}\|x^* - x_k\|^2$, and check,

$$\begin{aligned}\frac{\tilde{E}_{k+1}-\tilde{E}_k}{\delta} = -\left\langle \frac{x_{k+1}-x_k}{\delta}, x^* - x_k\right\rangle + \varepsilon_k^1 &\overset{(2.3)}{=} \langle \nabla f(x_k), x^* - x_k\rangle + \varepsilon_k^1 \\ &\overset{(A.6)}{\leq} -\mu\frac{1}{2}\|x^* - x_k\|^2 + \varepsilon_k^2 \leq -\mu\tilde{E}_k,\end{aligned}$$

where $\varepsilon_k^1 = \langle \nabla f(x_k), x_k - x_{k+1}\rangle - \frac{1}{2\delta}\|x_{k+1}-x_k\|^2$ and $\varepsilon_k^2 = f(x^*) - f(x_k) + \varepsilon_k^1$. The last upper bound $\varepsilon_k^2 \leq 0$ follows from the $(1/\delta)$-smoothness assumption on f (A.13). Therefore, the recurrence $\tilde{E}_{k+1} - \tilde{E}_k = -\delta\mu\tilde{E}_k$ allows us to conclude a $O(e^{-\mu\delta k}) \approx O((1-(\mu\delta))^k) = O((1-\kappa^{-1})^k)$ convergence rate,

$$\frac{1}{2}\|x^* - x_k\|^2 \leq (1-\kappa^{-1})^k E_0.$$

Proximal Method *For PM, the scaled distance function,*

$$E_k = (1+\mu\delta)^k\frac{1}{2}\|x^* - x_k\|^2,$$

is a Lyapunov function. We check,

$$\begin{aligned}\frac{\tilde{E}_{k+1}-\tilde{E}_k}{\delta} = -\left\langle \frac{x_{k+1}-x_k}{\delta}, x^* - x_{k+1}\right\rangle + \varepsilon_k^1 &\overset{(2.4)}{=} \langle \nabla f(x_{k+1}), x^* - x_{k+1}\rangle + \varepsilon_k^1 \\ &\overset{(A.6)}{\leq} -\mu\frac{1}{2}\|x^* - x_{k+1}\|^2 = -\mu\tilde{E}_{k+1},\end{aligned}$$

where $\varepsilon_k^1 = -\frac{1}{2\delta}\|x_{k+1}-x_k\|^2$ is negative. Therefore, the recurrence $\tilde{E}_{k+1} - \tilde{E}_k \leq -\mu\delta\tilde{E}_{k+1}$ allows us to conclude a $O(e^{-\mu\delta k}) \approx O((1+\mu\delta)^{-k})$ convergence rate,

$$f(x_k) - f(x^*) \leq (1+\mu\delta)^{-k}E_0.$$

Tighter Bound: *If we assume the f is L-smooth in continuous time, we obtain a tighter bound for GF when f is μ-strongly convex using the bound ($\mu \leq L$),*

$$\langle \nabla f(X_t), x^* - X_t\rangle \leq -\frac{\mu L}{\mu+L}\|x^* - X_t\|^2 - \frac{1}{\mu+L}\|\nabla f(X_t)\|^2. \tag{B.4}$$

We provide a proof of this bound in the Appendix B.1.4. Using (B.4), it follows that,

$$\frac{d}{dt}\tilde{\mathcal{E}}_t = -\langle \dot{X}_t, x^* - X_t\rangle \overset{(2.2)}{=} \langle \nabla f(X_t), x^* - X_t\rangle \overset{(B.4)}{\leq} -\frac{2\mu L}{\mu + L}\tilde{\mathcal{E}}_t$$

for $\tilde{\mathcal{E}}_t = \frac{1}{2}\|x^* - X_t\|^2$. *Thus,*

$$\mathcal{E}_t = e^{\frac{2\mu L}{\mu+L}t}\frac{1}{2}\|x^* - X_t\|^2,$$

is a Lyapunov function; we can subsequently conclude the upper bound,

$$\frac{1}{2}\|x^* - X_t\|^2 \leq e^{-\frac{2\mu L}{\mu+L}t}\mathcal{E}_0,$$

and an $O(e^{-\frac{2\mu L}{\mu+L}t})$ *convergence rate. In addition, smoothness can be used to conclude a tighter bound on the convergence of the optimality gap using this tighter bound,*

$$f(X_t) - f(x^*) \leq \frac{L}{2}\|x^* - X_t\|^2 \leq e^{-\frac{2\mu L}{\mu+L}t}\frac{L}{2}\|x^* - X_0\|^2.$$

Similar improved bounds can be obtained for GD and PM under the same conditions.

Gradient Descent *Define* $\tilde{E}_k = \frac{1}{2}\|x^* - x_k\|^2$. *The following bound holds for GD,*

$$\frac{\tilde{E}_{k+1} - \tilde{E}_k}{\delta} = -\left\langle \frac{x_{k+1} - x_k}{\delta}, x^* - x_k\right\rangle + \varepsilon_k^1 \overset{(2.3)}{=} \langle \nabla f(x_k), x^* - x_k\rangle + \varepsilon_k^1 \overset{(B.4)}{=} -\frac{2\mu L}{\mu + L}\tilde{E}_k + \varepsilon_k^2,$$

where $\varepsilon_k^1 = \frac{\delta}{2}\|\nabla f(x_k)\|^2$ *and* $\varepsilon_k^2 = -\left(\frac{1}{\mu+L} - \frac{\delta}{2}\right)\|\nabla f(x_k)\|^2$. *If we take* $0 < \delta < \frac{2}{\mu+L}$, *then* $\varepsilon_k^2 \leq 0$, *and we obtain the recursion* $\tilde{E}_{k+1} \leq \left(1 - \frac{2\mu L}{\mu+L}\delta\right)\tilde{E}_k$. *Let* $\delta = 2/(\mu + L)$. *Then,*

$$E_k = \left(1 - \frac{2\mu L}{\mu + L}\delta\right)^{-k}\frac{1}{2}\|x^* - x_k\|^2 = \left(\frac{\kappa - 1}{\kappa + 1}\right)^{-2k}\frac{1}{2}\|x^* - x_k\|^2$$

is a Lyapunov function, from which we conclude,

$$\frac{1}{2}\|x^* - x_k\|^2 \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^{2k} E_0.$$

Using the Lyapunov argument as well as the smoothness condition (A.13), we can also conclude a stronger bound on the optimality gap,

$$f(x_k) - f(x^*) \leq \frac{L}{2}\|x^* - x_k\|^2 \leq e^{-\frac{2\mu L}{\mu+L}\delta k}\frac{L}{2}\|x^* - x_0\|^2.$$

Proximal Method *Define* $\tilde{E}_k = \frac{1}{2}\|x^* - x_k\|^2$. *The following bound holds for the PM,*

$$\frac{\tilde{E}_{k+1} - \tilde{E}_k}{\delta} = -\left\langle \frac{x_{k+1} - x_k}{\delta}, x^* - x_{k+1} \right\rangle + \varepsilon_k^1 \overset{(2.4)}{=} \langle \nabla f(x_{k+1}), x^* - x_{k+1} \rangle + \varepsilon_k^1 \overset{(B.4)}{=} -\frac{2\mu L}{\mu + L}\tilde{E}_{k+1} + \varepsilon_k^2$$

where both of the errors $\varepsilon_k^1 = -\frac{\delta}{2}\|\nabla f(x_{k+1})\|^2$ *and* $\varepsilon_k^2 = -\left(\frac{\delta}{2} + \frac{1}{\mu+L}\right)\|\nabla f(x_{k+1})\|^2$ *are negative. This allows us to obtain the recursion* $\tilde{E}_{k+1} \leq \left(1 + \frac{2\mu L}{\mu+L}\delta\right)^{-1}\tilde{E}_k$ *for any* $\delta > 0$, *and that*

$$E_k = \left(1 + \frac{2\mu L}{\mu + L}\delta\right)^k \frac{1}{2}\|x^* - x_k\|^2,$$

is a Lyapunov function. Smoothness gives an improved upper bound on the function,

$$f(x_k) - f(x^*) \leq \frac{L}{2}\|x^* - x_k\|^2 \leq e^{-\frac{2\mu L}{\mu+L}\delta k}\frac{L}{2}\|x^* - x_0\|^2,$$

for the proximal method.

B.1.3 Summary

To summarize, we have presented several Lyapunov functions for the gradient flow equation (2.2), which provides a tool to conclude a non-asymptotic rate of convergence. When the objective function is differentiable, Lipschitz and $\mathcal{X} = \mathbb{R}^d$, *we can expect the function to converge to a critical point at the rate* $O(1/t)$. *If in addition* f *is convex, so that all local minima are global minimum and there are no saddle points, we can guarantee convergence of the optimality gap defined by a minimizer of* f. *If* f *is* μ*-strongly convex, i.e. the optimality gap is bounded below by* $O(-\frac{1}{2\mu}\|\nabla f(X_t)\|^2)$, *we can expect a much faster convergence rate of* $O(e^{-\mu t})$. *However, in discrete-time if* $\mu\delta := \kappa^{-1}$ *is too small,* δ *is the discretization set and a measure of the smoothness of* f *(*f *is* $(1/\delta)$*-smooth), then the difference between these two settings diminishes.*

For the proximal method we can think of the step-size δ *as a regularizer, which determines the trade off between minimizing the function* $f(x)$ *and keeping the point close to the current iterate* $\frac{1}{2\delta}\|x - x_k\|^2$. *The larger the step-size, the smaller the distance function, and the more we prioritize minimizing the function. This intuitively leads to a faster rate of convergence. However, unless* f *is somehow simple, it may be hard to solve the subproblem, and so using this method is often impractical. This accounts for why gradient descent is one of the most widely used algorithms in optimization.*

We now end by giving a brief description of what changes when $\mathcal{X}$ *is a compact, convex set, how to analyze proximal gradient descent, and the Lyapunov property is used to pick the step-size.*

Projections *Suppose $\mathcal{X} \subseteq \mathbb{R}^d$ is a convex, compact set. We can instead write the GF update (2.2) as*

$$\begin{aligned} \frac{d}{dt}Y_t &= -\nabla f(X_t) \\ X_t &= \Pi_{\mathcal{X}}(Y_t) \end{aligned} \tag{B.5}$$

where $\Pi_{\mathcal{X}}(x) = \arg\min_{y\in\mathcal{X}} \frac{1}{2}\|y-x\|^2$ is the projection operator onto the set $\mathcal{X}$.

We can similarly write the GD update (2.3) as

$$\begin{aligned} y_{k+1} &= x_k - \delta\nabla f(x_k) \\ x_{k+1} &= \Pi_{\mathcal{X}}(y_{k+1}) \end{aligned} \tag{B.6}$$

using the same projection operator. Furthermore, nearly the same arguments presented in this section follow from using the same Lyapunov functions; the only modification in each analysis involves using the property

$$-\left\langle \frac{x_{k+1}-x}{\delta}, \frac{x_{k+1}-y_{k+1}}{\delta} \right\rangle \leq 0, \forall x \in \mathcal{X} \tag{B.7}$$

for the Lyapunov arguments which entail showing the descent property, and the property

$$\|x^* - x_{k+1}\|^2 \leq \|x^* - y_{k+1}\|^2,$$

for Lyapunov arguments involving the metric, at the beginning of each analysis. Both inequalities follow from properties of the projection operator (A.28). Each analysis then proceeds accordingly with the same Lyapunov arguments. To provide an explicit example, for argument (2.8), we have the simple modification,

$$\frac{f(x_{k+1})-f(x_k)}{\delta} \leq \frac{f(y_{k+1})-f(x_k)}{\delta} \leq \frac{2-\delta L}{2}\left\langle \nabla f(x_k), \frac{y_{k+1}-x_k}{\delta} \right\rangle = -\frac{2-\delta L}{2}\|\nabla f(x_k)\|^2$$

where the first inequality uses the convexity of f and property (B.7) with $x = y_{k+2}$ to upper-bound $\frac{f(x_{k+1})-f(y_{k+1})}{\delta} \leq 0$.

Proximal gradient descent *Suppose we can decompose the objective function into two components, $f = f_1 + f_2$, where one of them is easy to optimize over. The forward-backward splitting method (also called the proximal gradient method), is obtained by applying the forward-Euler (1.6) discretization to f_1 and the backward-Euler (1.5) discretization to f_2, the part that is easy to optimize,*

$$y_{k+1} = x_k - \delta\nabla f_2(x_k) \tag{B.8a}$$

$$x_{k+1} = Prox_{f_1}(y_{k+1}). \tag{B.8b}$$

Recall, $Prox_{\delta f_1}(x) = \arg\min_{x\in\mathcal{X}} \left\{ f_1(x) + \frac{1}{2\delta}\|x-y\|^2 \right\}$. *We can write the variational condition for the combined update* (B.8) *as,*

$$\frac{x_{k+1}-x_k}{\delta} = -\nabla f_1(x_{k+1}) - \nabla f_2(x_k).$$

Lyapunov arguments for the FB-splitting method can be obtained by combining Lyapunov functions for the proximal method and gradient descent appropriately. This follows from the fact that (1) the two vector fields are additive, and (2) both discretizations use the same Lyapunov functions. For an excellent monograph on proximal algorithms, see [53]

Choosing δ using Lypaunov functions *We showed that choosing the optimal step-size δ for gradient descent required knowing the smoothness of the function, and sometimes the strong convexity parameter μ. Often we know neither of these. Most ways of searching for good step-sizes for GD are constructed around the idea of making the function value $\mathcal{E}_t = f(X_t)$ (i.e. the Lyapunov function $\mathcal{E}_t = f(X_t) - f(x^*)$ shifted by a constant $f(x_*)$) go down, given we know that it should be decreasing as function of time for any smooth function. For instance,* exact line search *is a technique which solves the subproblem,*

$$\min_{\delta>0} \frac{f(x_{k+1})-f(x_k)}{\delta} = \min_{\delta>0} \frac{1}{\delta}(f(x_k - \delta\nabla f(x_k)) - f(x_k))$$

or more simply,

$$\min_{\delta>0} f(x_{k+1}) - f(x_k) = \min_{\delta>0} f(x_k - \delta\nabla f(x_k)).$$

Often, however, this subproblem is too expensive to solve for practical applications. Other techniques for choosing step-sizes, such as the weak Wolfe conditions *and* backtracking line search *also use criterion formulated around the idea of making the function value go down.*

B.1.4 Tighter Bound

This proof follows [43, p. 2.1.12]. Define $\phi(x) = f(x) - \frac{\mu}{2}\|x\|^2$ and note $\nabla_x\phi(x) = \nabla f(x) - \mu x$. The smoothness of f, i.e. $\langle \nabla f(x) - \nabla f(y), x-y\rangle \le L\|x-y\|^2$ implies $\langle \phi(x) - \phi(y), x-y\rangle \le (L-\mu)\|x-y\|^2$ (i.e. that ϕ is $L-\mu$-smooth.) This, in turn, implies $\langle \nabla f(x) - \nabla f(y), x-y\rangle\rangle \le \frac{1}{L-\mu}\|\nabla\phi(x) - \phi(y)\|^2$, which when expanded, is our condition (B.4)

B.2 Mirror Desent

B.2.1 Differentiable Function

The steepest descent dynamic is called natural gradient flow *(NGF),*

$$\dot{X}_t = \arg\min_{v\in\mathbb{R}^d} \left\{ \langle \nabla f(X_t), v\rangle + \frac{1}{2}\|v\|^2_{\nabla^2 h(X_t)} \right\} = -\nabla^2 h(X_t)^{-1}\nabla f(X_t). \tag{B.9a}$$

The optimality gap, $\mathcal{E}_t = f(X_t) - f(x^*)$ is a Lyapunov function for (B.9),

$$\frac{d}{dt}\mathcal{E}_t = \langle \nabla f(X_t), \dot{X}_t \rangle \overset{(B.9)}{=} -\|\nabla f(X_t)\|^2_{X_t^*} \leq 0. \tag{B.10}$$

If in addition, f satisfies the PL condition (2.11),

$$\frac{d}{dt}\mathcal{E}_t = \langle \nabla f(X_t), \dot{X}_t \rangle \overset{(B.9)}{=} -\|\nabla f(X_t)\|^2_{X_t^*} \overset{(2.11)}{\leq} -2\mu(f(X_t) - f(x^*)),$$

we conclude

$$\mathcal{E}_t = e^{2\mu t}(f(X_t) - f(x^*))$$

is a Lyapunov function and an $O(e^{-\mu t})$ convergence rate. A similar statement can be made for natural gradient descent (NGD), the forward-Euler (1.6) discretization of natural gradient flow.

Natural Gradient Descent *Natural gradient descent [3],*

$$\frac{x_{k+1} - x_k}{\delta} = -\nabla^2 h(x_k)^{-1} \nabla f(x_k),$$

is also a steepest descent flow as long as f is $L = (1/\delta)$-smooth with respect the Hessian metric: $D_f(x, y) \leq \frac{1}{\delta}\|x - y\|_x$. If so, the discrete optimality gap, $E_k = f(x_k) - f(x^*)$, is a Lyapunov function for NGD:

$$\frac{E_{k+1} - E_k}{\delta} \leq \frac{2 - \delta L}{2} \left\langle \nabla f(x_k), \frac{x_{k+1} - x_k}{\delta} \right\rangle = -\frac{2 - \delta L}{2} \|\nabla f(x_k)\|^2_{x_k^*} \leq 0. \tag{B.11}$$

If in addition, f satisfies the PL-condition (2.11),

$$\frac{E_{k+1} - E_k}{\delta} \overset{(B.11)}{\leq} -\frac{1}{2}\|\nabla f(x_k)\|^2_{x_k^*} \overset{(2.11)}{\leq} -\mu(f(x_k) - f(x^*)),$$

then $E_k = (1 - \mu\delta)^{-k}(f(x_k) - f(x^*)$ is a Lyapunov function, and we can conclude a matching $O(e^{-\mu\delta k})$ convergence rate.

B.2.2 Convex Functions

Descent Property *MF and NGF are equivalent dynamics. Therefore, we can combine the descent property (B.10) and Lyapunov function (2.16) to conclude*

$$\mathcal{E}_t = D_h(x^*, X_t) + t(f(X_t) - f(x^*)) \tag{B.12}$$

is a Lyapunov function for MF/NGF. We check,

$$\frac{d}{dt}\mathcal{E}_t^{(\text{B.12})} = \frac{d}{dt}\mathcal{E}_t^{(2.13)} + t\frac{d}{dt}(f(X_t) - f(x^*)) \overset{(\text{B.10})}{\leq} -t\|\dot{X}_t\|_{X_t}^2 \leq 0.$$

Here, $\mathcal{E}_t^{(2.13)}$ represents the Lyapunov function defined by (2.13) and the second inequality uses the fact that we have shown this Lyapunov function is decreasing for MF (2.12). Therefore, if in addition, we can show our discretizations are descent methods, we can conclude a slightly stronger result. We now remark that the descent property can be shown for the BPM (2.17); subsequently, we can establish

$$E_k = D_h(x^*, x_k) + \delta k(f(x_k) - f(x^*)) \tag{B.13}$$

is a Lyapunov function. We check,

$$\frac{E_{k+1}^{(\text{B.13})} - E_k^{(\text{B.13})}}{\delta} \leq \frac{E_{k+1}^{(2.16)} - E_k^{(2.16)}}{\delta} + \delta k\frac{f(x_{k+1}) - f(x_k)}{\delta} \leq -\delta k\frac{1}{\delta^2}D_h(x_{k+1}, x_k) \leq 0,$$

where the second inequality follows because x_{k+1} satisfies $f(x_{k+1}) + \frac{1}{\delta}D_h(x_{k+1}, x_k) \leq f(x_k) + \frac{1}{\delta}D_h(x_k, x_k)$; we can therefore obtain the tighter convergence bound $f(x_k) - f(x^*) \leq E_0/\delta k$. Unfortunately, it is unclear whether the descent property holds for mirror descent without stronger conditions on the geometry h.

B.2.3 Strongly Convex Functions

The distance function, $\tilde{\mathcal{E}}_t = D_h(x^*, X_t)$, is a Lyapunov function for NGF/MF, when f is μ-strongly convex with respect to h (A.7), $D_f(x^*, X_t) \leq -\mu D_h(x^*, X_t)$. We check,

$$\begin{aligned}\frac{d}{dt}\tilde{\mathcal{E}}_t = -\left\langle \frac{d}{dt}\nabla h(X_t), x^* - X_t\right\rangle &\overset{(2.12)}{=} \langle \nabla f(X_t), x^* - X_t\rangle \\ &\overset{(\text{A.7})}{\leq} -\mu D_h(x^*, X_t) = -\mu\tilde{\mathcal{E}}_t.\end{aligned}$$

Notice that if f is just convex, the distance function $\tilde{\mathcal{E}}_t = D_h(x^*, X_t)$ is a Lyapunov function. The addition of the strong convexity condition allows us to conclude a rate of convergence; in particular, from the recurrence $\frac{d}{dt}\tilde{\mathcal{E}}_t \leq -\mu\tilde{\mathcal{E}}_t$, we can conclude

$$\mathcal{E}_t = e^{\mu t}D_h(x^*, X_t) \tag{B.14}$$

is also a Lyapunov function, as well as the convergence bound $D_h(x^*, X_t) \leq O(e^{-\mu t})$.

A similar statement can be made for mirror descent and the BPM.

Mirror Descent *When f is μ-strongly convex and $(1/\delta)$-smooth with respect to h (A.7), $\tilde{E}_k = D_h(x^*, x_k)$, is a Lyapunov function for mirror descent,*

$$\frac{\tilde{E}_{k+1} - \tilde{E}_k}{\delta} = -\left\langle \frac{\nabla h(x_{k+1}) - \nabla h(x_k)}{\delta}, x^* - x_k \right\rangle + \varepsilon_k^1 \overset{(2.12)}{=} -\langle \nabla f(x_k), x^* - x_k \rangle + \varepsilon_k^1$$
$$\leq -\mu D_h(x^*, x_k) + \varepsilon_k^2 \leq -\mu \tilde{E}_k.$$

Here, $\varepsilon_k^1 = \langle \nabla f(x_k), x_k - x_{k+1} \rangle - \frac{1}{\delta} D_h(x_{k+1}, x_k)$ and $\varepsilon_k^2 = f(x^*) - f(x_{k+1}) + D_f(x_{k+1}, x_k) - \frac{1}{\delta} D_h(x_{k+1}, x_k)$. The first inequality uses strong convexity and the second uses smoothness to upper bound the final error. From the recursion $\tilde{E}_{k+1} \leq (1 - \mu\delta)\tilde{E}_k \leq (1 - \mu\delta)^k \tilde{E}_0$, we can conclude

$$E_k = (1 - \mu\delta)^{-k} D_h(x^*, x_k),$$

is a Lyapunov function, as well as the convergence bound, $D_h(x^*, x_k) \leq O(e^{-\mu\delta k})$.

Bregman Proximal Minimization *When f is μ-strongly convex with respect to h (A.7), $\tilde{E}_k = D_h(x^*, x_k)$, is a Lyapunov function for BPM,*

$$\frac{\tilde{E}_{k+1} - \tilde{E}_k}{\delta} = -\left\langle \frac{\nabla h(x_{k+1}) - \nabla h(x_k)}{\delta}, x^* - x_{k+1} \right\rangle + \varepsilon_k^1 \overset{(2.17)}{=} -\langle \nabla f(x_{k+1}), x^* - x_{k+1} \rangle + \varepsilon_k^1$$
$$\leq -\mu D_h(x^*, x_{k+1}) \leq -\mu E_{k+1}.$$

where the error $\varepsilon_k^1 = -\frac{1}{\delta} D_h(x_{k+1}, x_k)$. From the recursion $\tilde{E}_{k+1} \leq (1 + \mu\delta)^{-1} \tilde{E}_k \leq (1 + \mu\delta)^{-k} \tilde{E}_0$, we can conclude

$$E_k = (1 + \mu\delta)^k D_h(x^*, x_k),$$

is a Lyapunov function, as well as the convergence bound, $D_h(x^*, x_k) \leq O(e^{-\mu\delta k})$.

B.2.4 Summary

We end this section by giving brief descriptions of the relationship between NGF and MF, what changes when $\mathcal{X}$ is a compact, convex set, how to analyze proximal mirror descent, and connections between mirror descent and information geometry.

Mirror Maps *To summarize, on a Hessian Riemannian manifold, MF is the push-forward of natural gradient flow NGF under the mapping $\phi = \nabla h$,* and both are gradient flows. *Furthermore, this property, that flows on both spaces are gradient flows, is* unique *to Riemannian manifolds with a Hessian metric structure (see Theorem 3.1 of [2]). To illustrate this property, we demonstrate how the gradient flow changes as we map between these two spaces.*

Consider a function $f : \tilde{\mathcal{X}} \to \mathbb{R}$ defined on the set $\tilde{\mathcal{X}}$ and a smooth bijective map $\phi : \tilde{\mathcal{X}} \to \tilde{\mathcal{Y}}$. The push-forward of f function under the map leads to a new objective function $\tilde{f} : \tilde{\mathcal{Y}} \to \mathbb{R}$

defined on $\tilde{\mathcal{Y}}$*, given by* $\tilde{f} = f \circ \phi^{-1}$*. We compute the gradient of* $\tilde{f}$ *at a point* $y = \phi(x)$ *as* $\nabla \tilde{f}(y) = \partial_z \tilde{f}(z)|_{z=y} = \partial_z (f \circ \phi^{-1})(z)|_{z=y} = J_{\phi^{-1}}(y) \partial_z f(\phi^{-1}(z))|_{z=y} = J_{\phi^{-1}}(y) \nabla f(\phi^{-1}(y))$*, where* $J_{\phi^{-1}}(y)$ *is the Jacobian (partial derivatives) of* $\phi^{-1}(z)$ *at* $z = y$*, represented as a matrix. We compute how the general metric* $\mathrm{g}(x)$ *on* $\mathcal{X}$ *changes at a point, where we eventually make the choice* $\mathrm{g} = \nabla^2 h$*. The pullback metric of* g*, which we denote* $\tilde{\mathrm{g}} = \phi^{-1}\mathrm{g}$ *at a point* $y = \phi(x)$*, is given by* $\tilde{\mathrm{g}}(y) = J_{\phi^{-1}}(y)^\top (\mathrm{g} \circ \phi^{-1})(y) J_{\phi^{-1}}(y)$*. Putting the pieces together, we can calculate natural gradient flow on* $\tilde{\mathcal{Y}}$ *as* $\dot{Y}_t = -\tilde{\mathrm{g}}(Y_t)^{-1} \nabla \tilde{f}(Y_t) = -J_{\phi^{-1}}(Y_t)^{-1} \mathrm{g}(\phi^{-1}(Y_t))^{-1} \nabla f(\phi^{-1}(Y_t))$*.*

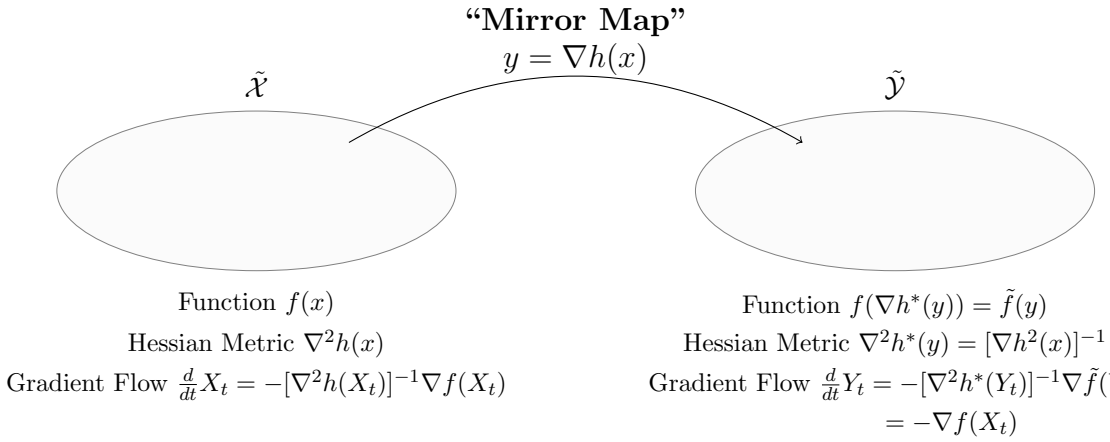

Figure B.1: The mirror map represents the duality relationship between MF and NGF.

If we take $\mathrm{g} = \nabla^2 h$ *and* $\phi^{-1} = \nabla h^*$*, where* $h^* : \tilde{\mathcal{X}}^* \to \tilde{\mathcal{X}} \to$ *is the* Legendre dual function *defined on a the dual space* $\tilde{\mathcal{X}}^*$*, then we obtain the identities,*

$$\nabla h(\nabla h^*(y)) = y \tag{B.15}$$

and

$$\nabla^2 h^*(y) \nabla^2 h(\nabla h^*(y)) = I. \tag{B.16}$$

Using these identities, we can write the gradient flow on $\tilde{\mathcal{Y}} = \tilde{\mathcal{X}}^*$ *as* $\frac{d}{dt} Y_t = -\nabla f(\nabla h^*(Y_t)) = -\nabla f(X_t)$*.*

Projections *We adopt the setting described at the beginning of the subsection, where* $\mathcal{X} \subseteq \mathbb{R}^d$ *is a convex, compact set. We can write the MF* (2.12) *as the system of differential equations,*

$$\frac{d}{dt} Y_t = -\nabla f(X_t) \tag{B.17a}$$

$$X_t \in \Pi_{\mathcal{X} \cap \tilde{\mathcal{X}}}(\nabla h^*(Y_t)). \tag{B.17b}$$

Here,

$$\Pi_{\mathcal{X}\cap\tilde{\mathcal{X}}}(x) := \arg\min_{y\in\mathcal{X}\cap\tilde{\mathcal{X}}} D_h(y,x) \tag{B.18}$$

is a Bregman projection operator.

We can similarly write MD as,

$$y_{k+1} = \nabla h(x_k) - \delta\nabla f(x_k) \tag{B.19a}$$
$$x_{k+1} \in \Pi_{\mathcal{X}\cap\tilde{\mathcal{X}}}(\nabla h^*(y_{k+1})) \tag{B.19b}$$

where we use the same Bregman projection operator $\Pi_{\mathcal{X}}(x) = \arg\min_{y\in\mathcal{X}\cap\tilde{\mathcal{X}}} D_h(y,x)$*. A similar variational condition* $\nabla h(x_k) = y_k$ *is implied using the mirror map. For this modified algorithm, nearly all of the same Lyapunov arguments presented in this section follow using the same Lyapunov functions; the only modification in each analysis involves recognizing the following property of projection operator* (A.28)*,*

$$D_h(x^*, x_{k+1}) \le D_h(x^*, y_{k+1})$$

As a specific example, the following argument can be made for mirror descent where f *is* μ*-strongly convex with respect to* h (A.7)*, using* $\tilde{E}_k = D_h(x^*, x_k)$*:*

$$\frac{\tilde{E}_{k+1}-\tilde{E}_k}{\delta} \le -\left\langle \frac{\nabla h(y_{k+1})-\nabla h(x_k)}{\delta}, x^*-x_k\right\rangle + \varepsilon_k^1 \overset{\text{(B.19a)}}{=} \langle \nabla f(x_k), x^*-x_k\rangle + \varepsilon_k^1$$
$$\le -\mu D_h(x^*,x_k) + \varepsilon_k^2 \le -\mu\tilde{E}_k$$

Here, the first line uses the inequality $\frac{\tilde{E}_{k+1}-\tilde{E}_k}{\delta} \le \frac{1}{\delta}D_h(x^*,y_{k+1}) - \frac{1}{\delta}D_h(x^*,x_k)$*. The first error scales as* $\varepsilon_k^1 = -\langle(\nabla h(y_{k+1})-\nabla h(x_k))/\delta, x_k - y_{k+1}\rangle + \frac{1}{\delta}D_h(y_{k+1},x_k) = \langle\nabla f(x_k), x_k - y_{k+1}\rangle - \frac{1}{\delta}D_h(y_{k+1},x_k)$*. The second error scales as* $\varepsilon_k^2 = f(x^*) - f(y_{k+1}) - D_f(y_{k+1},x_k) - \frac{1}{\delta}D_h(y_{k+1},x_k)$*, which we can upper bound using the* $(1/\delta)$*- smoothness of* f *with respect to* h (A.14)*. In other words, its the same argument as the unconstrated case, where we replace* x_{k+1} *with* y_{k+1}*.*

A projection step can be similarly added to each update of the MP algorithm (2.19) *and the same Lyapunov functions used, similar to what we have just shown.*

Proximal Mirror Descent *Suppose we can decompose the objective function into two components,* $f = f_1 + f_2$ *where one of them is easy to optimize over. The forward-backward splitting method, i.e. proximal mirror descent, is obtained by applying the forward-Euler* (1.6) *discretization to* f_1 *and backward-Euler* (1.5) *discretization to* f_2*:*

$$y_{k+1} = \nabla h(x_k) - \delta\nabla f_2(x_k) \tag{B.20a}$$
$$x_{k+1} \in Prox_{\delta f_1}^{h,\tilde{\mathcal{X}}\cap\mathcal{X}}(\nabla h^*(y_{k+1})). \tag{B.20b}$$

Here, $Prox_{\delta f_1}^{h,\tilde{\mathcal{X}}\cap\mathcal{X}} = \arg\min_{x\in\tilde{\mathcal{X}}\cap\mathcal{X}}\{f_1(x)+\frac{1}{\delta}D_h(x,y)\}$ *is the Bregman proximal function. The update* (B.20) *satisfies the variational condition,*

$$\frac{\nabla h(x_{k+1})-\nabla h(x_k)}{\delta} = -\nabla f_2(x_k) - \nabla f_1(x_{k+1}).$$

Lyapunov arguments for this FB-splitting method can be obtained by combing the Lyapunov functions for the Bregman method and mirror descent, given the vector field is additive and the same Lyapunov functions can be used in both settings. We relax the differentiability of f_1 *in the next subsection.*

Relevant Citations *The Bregman Proximal Minimization has a long established history [11]. Mirror descent has also been discussed by many [39, 8]. The continuous time Lyapunov function* (2.13) *have been remarked on many times [39, 2, 75]; so too, has the dual relationship between MF and NGF [8, 61]. The connection between MF/MD and information geometry, game theory and theormodynamics has been extensively studied by many [23, 7, 22]. For a particularly nice survey on the connection between the replicator equation, evolutionary game theory, Nash equilibria and biology, see [7]*

B.3 Subgradients and Time Reparameterization

B.3.1 Strongly Convex Functions

We analyze dynamics (2.22) *in the setting where* f *is* μ*-strongly convex with respect to* h (A.7) *using the parameterized Lyapunov function* (B.14),

$$\mathcal{E}_{\tau_t} = e^{\mu\tau_t} D_h(x^*, Y_t). \tag{B.21}$$

Observe that,

$$\begin{aligned}
\frac{d}{dt}\mathcal{E}_{\tau_t} &= D_h(x^*,Y_t)\frac{d}{dt}e^{\mu\tau_t} + e^{\mu\tau_t}\frac{d}{dt}D_h(x^*,Y_t)\\
&= \mu\dot{\tau}_t e^{\mu\tau_t} D_h(x^*,Y_t) - e^{\mu\tau_t}\left\langle \frac{d}{dt}\nabla h(Y_t), x^*-Y_t\right\rangle\\
&= e^{\mu\tau_t}\dot{\tau}_t\left(\mu D_h(x^*,Y_t) + \left\langle G(Y_t,\dot{Y}_t), x^*-Y_t\right\rangle\right)\\
&\overset{(A.6)}{\leq} -\frac{1}{\mu}(f(X_t)-f(x^*))\frac{d}{dt}e^{\mu\tau_t}
\end{aligned} \tag{B.22}$$

From this argument, we obtain the upper bound,

$$D_h(x^*,Y_t) \leq \frac{e^{\mu\tau_0}D_h(x^*,Y_0)}{e^{\mu\tau_t}}$$

and an $O(e^{\mu\tau_t})$ convergence rate. Define $\hat{X}_t = \int_0^t X_s de^{\mu\tau_s}/e^{\mu\tau_t}$. By Jensen's $f(\hat{X}_t) \leq \int_0^t f(X_s) de^{\mu\tau_s}$. From (B.22) we can also conclude the upper bound,

$$f(\hat{X}_t) - f(x^*) \leq \frac{\mu \mathcal{E}_{\tau_0}}{e^{\mu\tau_t}},$$

as well as the fact that

$$\mathcal{E}_{\tau_t} = e^{\mu\tau_t} D_h(x^*, Y_t) + \frac{1}{\mu}\int_0^t f(Y_s) - f(x^*) de^{\mu\tau_s}$$

is a Lyapunov function.

Notice the difference between these two settings. In the convex case, the two scalings of time which appear explicitly in the dynamics and Lyapunov function are the tuple $(\tau_t, \dot{\tau}_t)$. The two scalings of time that appear in the dynamic and Lyapunov function in the strongly convex setting is $(e^{\mu\tau_t}, \dot{\tau}_t)$. Therefore, when we identify the first element in the tuple with the discrete sequence A_k, the approximation of the time derivative will differ as $(A_k, \frac{A_{k+1}-A_k}{\delta})$ and $(A_k, \frac{A_{k+1}-A_k}{\mu\delta A_{k+1}})$, respectively. In the latter case, we made the approximation $\dot{\tau}_t = \frac{d}{dt}e^{\mu\tau_t}/\mu e^{\mu\tau_t} \approx (A_{k+1} - A_k)/\mu\delta A_{k+1} := \alpha_k$. With this approximation, the same argument can be made for the scaled MD (2.24) and PM (2.27).

Mirror descent for non-smooth functions *To analyze mirror descent (2.24) when f is μ-strongly convex with respect to h (A.7) we use the scale-free Lyapunov function*

$$E_{A_k} = A_k D_h(x^*, y_k).$$

Observe that,

$$\begin{aligned}
\frac{E_{A_{k+1}} - E_{A_k}}{\delta} &= D_h(x^*, y_k)\frac{A_{k+1} - A_k}{\delta} + A_{k+1}\frac{D_h(x^*, y_{k+1}) - D_h(x^*, y_k)}{\delta} \\
&= A_{k+1}\alpha_k \mu D_h(x^*, y_k) - A_{k+1}\left\langle \frac{\nabla h(y_{k+1}) - \nabla h(y_k)}{\delta}, x^* - y_k \right\rangle + \varepsilon_k^1 \\
&= A_{k+1}\alpha_k(\mu D_h(x^*, y_k) - \langle g(y_k), x^* - y_k\rangle) + \varepsilon_k^1 \leq \varepsilon_k^2
\end{aligned} \tag{B.23}$$

where the first error scales as $\varepsilon_k^1 = A_{k+1}(\alpha_k\langle g(y_k), y_k - y_{k+1}\rangle - \frac{1}{\delta}D_h(y_{k+1}, y_k))$ and the second as, $\varepsilon_k^2 = A_{k+1}(\alpha_k\langle g(y_k), y_k - y_{k+1}\rangle - \frac{1}{\delta}D_h(y_{k+1}, y_k) - \alpha_k(f(y_k) - f(x^*)) \leq \varepsilon_k^1$. We can upper bound ε_k^2 using the same argument as in the convex case. We assume h is σ-strongly convex and apply Young's inequality (A.25) to obtain the bound $\varepsilon_k^2 \leq \varepsilon_k^1 \leq \frac{(A_{k+1}-A_k)^2}{2\mu^2\sigma\delta A_{k+1}}\|g(y_k)\|_*^2 - \frac{A_{k+1}-A_k}{\delta\mu}(f(y_k) - f(x^*)) := \varepsilon_k^3$. Assume $\|\partial f(y)\|_*^2 \leq G^2$ for all $y \in \mathcal{X}$ and some finite constant G. Choosing $A_k = (k+1)k$ and $\delta = 1$ so that $\alpha_k = \frac{2}{\delta\mu(k+2)}$ and $\tilde{\alpha}_k = \alpha_k A_{k+1} = \frac{A_{k+1}-A_k}{\delta\mu} = 2(k+1)/\delta\mu$, we obtain the upper bound

$$D_h(x^*, y_k) \leq \frac{A_0 D_h(x^*, y_0) + \delta\frac{1}{2\sigma}\sum_{s=0}^k \frac{\tilde{\alpha}_k^2}{A_{k+1}}G^2}{A_k} \tag{B.24}$$

and an $O(1/k)$ convergence rate. Define $\hat{y}_k = \sum_{s=0}^k y_s\alpha_s/A_k$. By Jensen's $f(\hat{x}_k) \leq \sum_{s=0}^k f(y_s)\alpha_s$. From (B.23) we can also conclude the upper bound

$$f(\hat{y}_k) - f(x^*) \leq \frac{\mu E_0 + \delta\frac{1}{2\sigma}\sum_{s=0}^k \frac{\tilde{\alpha}_k^2}{A_{k+1}}G^2}{A_k}$$

as well as the fact that

$$E_{A_k} = A_k D_h(x^*, y_k) + \frac{1}{\mu}\sum_{s=0}^{k-1}(f(x_s) - f(x^*))\frac{A_{s+1} - A_s}{\delta}\delta,$$

is a Lyapunov function.

Holder Continuous Gradients *If f is Holder-smooth (A.15), the following Lyapunov function,*

$$E_{A_k} = D_h(x^*, y_k) + \sum_{s=0}^{k}(f(y_s) - f(x^*))\frac{A_{s+1} - A_s}{\delta}.$$

for mirror descent (2.22) will full gradients. Using the analysis in the unsmooth case, it is to check

$$\frac{E_{A_{k+1}} - E_{A_k}}{\delta} = -D_f(x^*, y_k)\alpha_k + \varepsilon_k^1$$

where the error scales as $\varepsilon_k^1 = \alpha_k(f(y_{k+1}) - f(y_k) + \langle\nabla f(y_k), y_k - y_{k+1}\rangle) - \frac{1}{\delta}D_h(y_{k+1}, y_k)$. Using the Holder continuity of the gradients, along with the σ-strong convexity of f, we obtain the upper bound $\varepsilon_k^1 = \frac{L}{1+\nu}\alpha_k\|y_{k+1} - y_k\|^{1+\nu} - \frac{\sigma}{\delta}\|y_{k+1} - y_k\|^2$. We upper bound the error using Young's inequality (A.25) $\frac{1}{1+\nu}t^{1+\nu} \leq \frac{1}{2s}t^2 + \frac{1-\nu}{1+\nu}s^{\frac{1+\nu}{1-\nu}}$, with $t = \|y_{k+1} - y_k\|$ and $s = \delta\alpha_k L/\sigma$ provides the upper bound $\varepsilon_k^1 \leq \frac{1-\nu}{1+\nu}\frac{1}{2}\alpha_k^{\frac{2}{1-\nu}}L(\sigma L/\sigma)^{\frac{1+\nu}{1-\nu}}$. The choice $\alpha_k = D_h(x^*, x_0)/k^{\frac{1+\nu}{2}}$ leads to an $O(k^{-\frac{1+\nu}{2}})$ convergence rate.

B.4 Accelerated Mirror Prox

$$\begin{aligned}\frac{E_{k+1} - E_k}{\delta} &= -\left\langle\frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta}, x^* - x_{k+1}\right\rangle + \alpha_k(f(x_{k+1}) - f(x^*)) + \varepsilon_k^1 \\ &\overset{(2.55d)}{=} D_f(x_{k+1}, x^*)\alpha_k + \varepsilon_k^1\end{aligned}$$

where the error scales as,

$$\begin{aligned}\varepsilon_k^1 &= A_k \frac{f(x_{k+1}) - f(x_k)}{\delta} - \left\langle \frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta}, x_{k+1} - z_{k+1} \right\rangle - \frac{1}{\delta} D_h(z_{k+1}, z_k) \\ &\overset{\substack{(2.55d)\\(A.27)}}{=} A_k \frac{f(x_{k+1}) - f(x_k)}{\delta} + \alpha_k \langle \nabla f(x_{k+1}), x_{k+1} - z_{k+1} \rangle - \frac{1}{\delta} D_h(z_{k+1}, z'_{k+1}) \\ &\quad - \frac{1}{\delta} D_h(z'_{k+1}, z_k) - \left\langle \frac{\nabla h(z'_{k+1}) - \nabla h(z_k)}{\delta}, z_{k+1} - z'_{k+1} \right\rangle\end{aligned}$$

Using convexity, we can further upper-bound the error as folows,

$$\begin{aligned}\varepsilon_k^1 &\overset{(2.55b)}{\leq} A_{k+1} \left\langle \nabla f(x_{k+1}), \frac{x_{k+1} - x_k}{\delta} \right\rangle + \alpha_k \langle \nabla f(x_{k+1}), x_k - z_{k+1} \rangle - \frac{1}{\delta} D_h(z_{k+1}, z'_{k+1}) \\ &\quad - \frac{1}{\delta} D_h(z'_{k+1}, z_k) + \alpha_k \langle \nabla f(x'_{k+1}), z_{k+1} - z'_{k+1} \rangle \\ &\overset{(2.55c)}{=} \alpha_k \langle \nabla f(x_{k+1}) - \nabla f(x'_{k+1}), z'_{k+1} - z_{k+1} \rangle - \frac{1}{\delta} D_h(z_{k+1}, z'_{k+1}) - \frac{1}{\delta} D_h(z'_{k+1}, z_k)\end{aligned}$$

Using the $(1/\epsilon)$-smoothness of f, Cauchy-Schwartz (A.26) and the identity $\frac{x_{k+1} - x'_{k+1}}{\delta} = \tau_k(z'_{k+1} - z_k)$, the inequality $\alpha_k \langle \nabla f(x_{k+1}) - \nabla f(x'_{k+1}), z'_{k+1} - z_{k+1} \rangle \leq \delta \frac{\alpha_k^2}{A_{k+1}\epsilon} \|z'_{k+1} - z_{k+1}\| \|z'_{k+1} - z_k\|$ and the σ-strong convexity of h gives the remaining upper bound (2.56).

B.5 Dynamics

B.5.1 Proof of Proposition 2.2.1

We compute the Euler-Lagrange equation for the second Bregman Lagrangian (2.39). Denote $z = x + e^{-\alpha_t}\dot{x}$. The partial derivatives of the Bregman Lagrangian can be written,

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial v}(X_t, \dot{X}_t, t) &= \mu e^{\beta_t + \gamma_t} \left(\nabla h(Z_t) - \nabla h(X_t) \right) \\ \frac{\partial \mathcal{L}}{\partial x}(X_t, \dot{X}_t, t) &= \mu e^{\alpha_t} \frac{\partial \mathcal{L}}{\partial v}(X_t, \dot{X}_t, t) - \mu e^{\beta_t + \gamma_t} \frac{d}{dt} \nabla h(X_t) - e^{\alpha_t + \beta_t + \gamma_t} \nabla f(X_t).\end{aligned}$$

We also compute the time derivative of the momentum $p = \frac{\partial \mathcal{L}}{\partial v}(X_t, \dot{X}_t, t)$,

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial v}(X_t, \dot{X}_t, t) = (\dot{\beta}_t + \dot{\gamma}_t) \frac{\partial \mathcal{L}}{\partial v}(X_t, \dot{X}_t, t) + \mu e^{\beta_t + \gamma_t} \frac{d}{dt} \nabla h(Z_t) - \mu e^{\beta_t + \gamma_t} \frac{d}{dt} \nabla h(X_t).$$

The terms involving $\frac{d}{dt} \nabla h(X)$ cancel and the terms involving the momentum will simplify under the scaling condition (2.37a) when computing the Euler-Lagrange equation $\frac{\partial \mathcal{L}}{\partial x}(X_t, \dot{X}_t, t) = \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial v}(X_t, \dot{X}_t, t)$. Compactly, the Euler-Lagrange equation can be written

$$\frac{d}{dt} \mu \nabla h(Z_t) = -\dot{\beta}_t \mu \left(\nabla h(Z_t) - \nabla h(X_t) \right) - e^{\alpha_t} \nabla f(x).$$

Remark *It interesting to compare with the partial derivatives of the first Bregman Lagrangian* (A.2)*,*

$$\frac{\partial \mathcal{L}}{\partial v}(X_t, \dot{X}_t, t) = e^{\gamma_t}\left(\nabla h(Z_t) - \nabla h(X_t)\right)$$
$$\frac{\partial \mathcal{L}}{\partial x}(X_t, \dot{X}_t, t) = e^{\alpha_t}\frac{\partial \mathcal{L}}{\partial v}(X_t, \dot{X}_t, t) - e^{\gamma_t}\frac{d}{dt}\nabla h(X_t) - e^{\alpha_t+\beta_t+\gamma_t}\nabla f(X_t),$$

as well as the derivative of the momentum,

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial v}(X_t, \dot{X}_t, t) = \dot{\gamma}_t\frac{\partial \mathcal{L}}{\partial v}(X_t, \dot{X}_t, t) + e^{\gamma_t}\frac{d}{dt}\nabla h(Z_t) - e^{\gamma_t}\frac{d}{dt}\nabla h(X_t).$$

For Lagrangian (A.2)*, not only do the terms involving* $\frac{d}{dt}\nabla h(X)$ *cancel when computing the Euler-Lagrange equation, but the ideal scaling will also force the terms involving the momentum to cancel as well.*

B.5.2 Hamiltonian Systems

Bregman Hamiltonian. *One way to understand a Lagrangian is to study its Hamiltonian, which is the Legendre conjugate (dual function) of the Lagrangian. Typically, when the Lagrangian takes the form of the difference between kinetic and potential energy, the Hamiltonian is the sum of the kinetic and potential energy. The Hamiltonian is often easier to study than the Lagrangian, since its second-order Euler-Lagrangian equation is transformed into a pair of first-order equations. In our case, the Hamiltonian corresponding to the Bregman Lagrangians* (2.36) *and are the following* Bregman Hamiltonians*,*

$$\mathcal{H}(x, p, t) = e^{\alpha_t+\gamma_t}\left(D_{h^*}\left(\nabla h(x) + e^{-\gamma_t}p,\, \nabla h(x)\right) + e^{\beta_t}f(x)\right), \tag{B.26}$$

and

$$\mathcal{H}(x, p, t) = e^{\alpha_t+\beta+\gamma_t}\left(\mu D_{h^*}\left(\nabla h(x) + \frac{1}{\mu}e^{-(\beta_t+\gamma_t)}p,\, \nabla h(x)\right) + f(x)\right), \tag{B.27}$$

respectively. These, indeed, have the form of the sum of the kinetic and potential energy. Here the kinetic energy is measured using the Bregman divergence of h^**, which is the convex dual function of* h*.*

Calculating the Hamiltonian *In this section we define and compute the Bregman Hamiltonian corresponding to the Bregman Lagrangian. In general, given a Lagrangian* $\mathcal{L}(x, v, t)$*, its Hamiltonian is defined by*

$$\mathcal{H}(x, p, t) = \langle p, v\rangle - \mathcal{L}(x, v, t) \tag{B.28}$$

where $p = \frac{\partial \mathcal{L}}{\partial v}$ *is the momentum variable conjugate to position. For the Bregman Lagrangian* (2.36)*, the momentum variable is given by*

$$p = \frac{\partial \mathcal{L}}{\partial v} = e^{\gamma_t} \left(\nabla h(x + e^{-\alpha_t} v) - \nabla h(x) \right). \tag{B.29}$$

We can invert this equation to solve for the velocity v*,*

$$v \;=\; e^{\alpha_t} \left(\nabla h^*(\nabla h(x) + e^{-\gamma_t} p) - x \right), \tag{B.30}$$

where h^* *is the conjugate function to* h *(recall the definition in* (A.24)*), and we have used the property that* $\nabla h^* = [\nabla h]^{-1}$*. So for the first term in the definition* (B.28) *we have*

$$\langle p, v \rangle = e^{\alpha_t} \big\langle p,\ \nabla h^*(\nabla h(x) + e^{-\gamma_t} p) - x \big\rangle.$$

Next, we write the Bregman Lagrangian $\mathcal{L}(x, v, t)$ *in terms of* (x, p, t)*. We can directly substitute* (B.30) *to the definition* (2.36) *and calculate the result. Alternatively, we can use the property that the Bregman divergences of* h *and* h^* *satisfy* $D_h(y, x) = D_{h^*}(\nabla h(x), \nabla h(y))$*. Therefore, we can write the Bregman Lagrangian* (2.36) *as*

$$\begin{aligned}
\mathcal{L}(x, v, t) &= e^{\alpha_t + \gamma_t} \left(D_{h^*} \left(\nabla h(x),\ \nabla h(x + e^{-\alpha_t} v) \right) - e^{\beta_t} f(x) \right) \\
&= e^{\alpha_t + \gamma_t} \left(D_{h^*} \left(\nabla h(x),\ \nabla h(x) + e^{-\gamma_t} p \right) - e^{\beta_t} f(x) \right) \\
&= e^{\alpha_t + \gamma_t} \left(h^*(\nabla h(x)) - h^*(\nabla h(x) + e^{-\gamma_t} p) + e^{-\gamma_t} \langle \nabla h^*(\nabla h(x) + e^{-\gamma_t} p), p \rangle - e^{\beta_t} f(x) \right),
\end{aligned}$$

where in the second step we have used the relation $\nabla h(x + e^{-\alpha_t} v) = \nabla h(x) + e^{-\gamma_t} p$ *from* (B.29)*, and in the last step we have expanded the Bregman divergence.*

Substituting these calculations into (B.28) *and simplifying, we get*

$$\mathcal{H}(x, p, t) \;=\; e^{\alpha_t + \gamma_t} \left(h^*(\nabla h(x) + e^{-\gamma_t} p) - h^*(\nabla h(x)) - \langle x,\, e^{-\gamma_t} p \rangle + e^{\beta_t} f(x) \right).$$

For the Bregman Langrangian (2.39)*, we can invert the equation for its momentum variable*

$$v \;=\; e^{\alpha_t} \left(\nabla h^*(\nabla h(x) + \frac{1}{\mu} e^{-(\gamma_t + \beta_t)} p) - x \right). \tag{B.31}$$

Now we can solve for the Hamiltonian

$$\begin{aligned}
\mathcal{H}(x, p, t) &= \langle p, v \rangle - \mathcal{L}(x, v, t) \\
&= e^{\alpha_t} \left\langle p, \nabla h^* \left(\frac{1}{\mu} e^{-(\gamma_t + \beta_t)} p + \nabla h(x) \right) - x \right\rangle \\
&\quad - e^{\alpha_t + \gamma_t + \beta_t} \left(\mu D_h(\nabla h^*(\nabla h(x) + \frac{1}{\mu} e^{-\gamma_t + \beta_t} p), x) - f(x) \right)
\end{aligned}$$

Expanding, we have

$$\begin{aligned}
\mathcal{H}(x,p,t) &= e^{\alpha_t}\left\langle p, \nabla h^*\left(\frac{1}{\mu}e^{-(\gamma_t+\beta_t)}p + \nabla h(x)\right) - x\right\rangle \\
&\quad - e^{\alpha_t+\gamma_t+\beta_t}\left(\mu(h^*(\nabla h(x)) - h^*\left(\nabla h(x) + \frac{1}{\mu}e^{-\gamma_t+\beta_t}p\right)\right) \\
&\quad - e^{\alpha_t}\left\langle \nabla h^*\left(\frac{1}{\mu}e^{-(\gamma_t+\beta_t)}p + \nabla h(x)\right), p\right\rangle - e^{\alpha_t+\gamma_t+\beta_t}f(x) \\
&= e^{\alpha_t+\gamma_t+\beta_t}\mu\left(h^*\left(\nabla h(x) + \frac{1}{\mu}e^{-(\gamma_t+\beta_t)}p\right) - h^*(\nabla h(x)) + \frac{1}{\mu}e^{-(\gamma_t+\beta_t)}\left\langle p, \nabla h^*(\nabla h(x))\right\rangle\right) \\
&\quad + e^{\alpha_t+\beta_t+\gamma_t}f(x) \\
&= e^{\alpha_t+\gamma_t+\beta_t}\left(\mu D_{h^*}\left(\nabla h(x) + \frac{1}{\mu}e^{-(\gamma_t+\beta_t)}p, \nabla h(x)\right) - f(x)\right)
\end{aligned}$$

Thus, our generalized Hamiltonian has the form

$$\mathcal{H}(x,p,t) = e^{\alpha_t+\gamma_t+\beta_t}\left(\mu D_{h^*}\left(\nabla h(x) + \frac{1}{\mu}e^{-(\gamma_t+\beta_t)}p, \nabla h(x)\right) - f(x)\right)$$

Hamiltonian equations of motion. *The second-order Euler-Lagrange equation of a Lagrangian can be equivalently written as a pair of first-order equations*

$$\frac{d}{dt}X_t = \frac{\partial\mathcal{H}}{\partial p}(X_t,P_t,t), \qquad \frac{d}{dt}P_t = -\frac{\partial\mathcal{H}}{\partial x}(X_t,P_t,t). \tag{B.32}$$

For the Bregman Hamiltonian (B.26), the equations of motion are given by

$$\frac{d}{dt}X_t = e^{\alpha_t}\left(\nabla h^*(\nabla h(X_t) + e^{-\gamma_t}P_t) - X_t\right) \tag{B.33a}$$

$$\frac{d}{dt}P_t = -e^{\alpha_t+\gamma_t}\nabla^2 h(X_t)\left(\nabla h^*(\nabla h(X_t) + e^{-\gamma_t}P_t) - X_t\right) + e^{\alpha_t}P_t - e^{\alpha_t+\beta_t+\gamma_t}\nabla f(X_t). \tag{B.33b}$$

Notice that the first equation (B.34a) recovers the definition of momentum (B.29). Furthermore, when $\dot\gamma_t = e^{\alpha_t}$, by substituting (B.34a) to (B.34b) we can write (B.34) as

$$\frac{d}{dt}\left\{\nabla h(X_t) + e^{-\gamma_t}P_t\right\} = \nabla^2 h(X_t)\,\dot X_t - \dot\gamma_t e^{-\gamma_t}P_t + e^{-\gamma_t}\dot P_t = -e^{\alpha_t+\beta_t}\nabla f(X_t).$$

Since $\nabla h(X_t)+e^{-\gamma_t}P_t = \nabla h(X_t+e^{-\alpha_t}\dot X_t)$ by (B.34a), this indeed recovers the Euler-Lagrange equation (2.38).

A Lyapunov function for the Hamiltonian equations of motion (B.34) is the following, which is simply the Lyapunov function (2.48) written in terms of (X_t,P_t,t),

$$\mathcal{E}_t = D_{h^*}\left(\nabla h(X_t) + e^{-\gamma_t}P_t,\ \nabla h(x^*)\right) + e^{\beta_t}(f(X_t) - f(x^*)).$$

For the Bregman Hamiltonian (B.27), the equations of motion are given by

$$\frac{d}{dt}X_t = e^{\alpha_t}\left(\nabla h^*(\nabla h(X_t) + \frac{1}{\mu}e^{-(\beta_t+\gamma_t)}P_t) - X_t\right) \tag{B.34a}$$

$$\frac{d}{dt}P_t = -e^{\alpha_t+\beta_t\gamma_t}\left(\mu\nabla^2 h(X_t)\left(\nabla h^*(\nabla h(X_t) + \frac{1}{\mu}e^{-(\beta_t+\gamma_t)}P_t) - X_t\right) - e^{-(\beta_t+\gamma_t)}P_t + \nabla f(X_t)\right). \tag{B.34b}$$

A Lyapunov function for the Hamiltonian equations of motion (B.34) is the following, which is simply the Lyapunov function (2.48) written in terms of (X_t, P_t, t),

$$\mathcal{E}_t = \mu D_{h^*}\left(\nabla h(X_t) + \frac{1}{\mu}e^{-(\beta_t+\gamma_t)}P_t,\, \nabla h(x^*)\right) + e^{\beta_t}(f(X_t) - f(x^*)).$$

The Hamiltonian formulation of the dynamics has appealing properties that seem worthy of further exploration. For example, Hamiltonian flow preserves volume in phase space (Liouville's theorem); this property has been used in the context of sampling to develop the technique of Hamiltonian Markov chain Monte-Carlo, and may also be useful to help us design better algorithms for optimization. Furthermore, the Hamilton-Jacobi-Bellman equation (which is a reformulation of the Hamiltonian dynamics) is a central object of study in the field of optimal control theory, and it would be interesting to study the Bregman Hamiltonian framework from that perspective.

B.6 Algorithms derived from (2.38)

We prove the following proposition, which is more general than proposition 2.2.3

Proposition B.6.1. *Assume that the distance-generating function h is σ-uniformly convex with respect to the p-th power of the norm ($p \geq 2$) (A.5) and the objective function is convex. Using only the updates (2.46a) and (2.46b), and using the Lyapunov function (2.48), we have the following bound:*

$$\frac{E_{k+1} - E_k}{\delta} \leq -\frac{A_{k+1} - A_k}{\delta} D_f(x_k, x^*) + \varepsilon_{k+1} \leq \varepsilon_{k+1}$$

where the error term scales as,

$$\varepsilon_{k+1} = \frac{p-1}{p}\sigma^{-\frac{1}{p-1}}\delta^{\frac{1}{p-1}}\alpha_k^{\frac{p}{p-1}}\|\nabla f(y_{k+1})\|^{\frac{p}{p-1}} + A_{k+1}\frac{f(y_{k+1}) - f(x_{k+1})}{\delta} \tag{B.35a}$$

If we use the updates (2.47a) and (2.47c) instead, the error term scales as,

$$\varepsilon_{k+1} = \frac{p-1}{p}\sigma^{-\frac{1}{p-1}}\delta^{\frac{1}{p-1}}\alpha_k^{\frac{p}{p-1}}\|\nabla f(y_{k+1})\|^{\frac{p}{p-1}} + A_{k+1}\left\langle \nabla f(y_{k+1}), \frac{y_{k+1} - x_{k+1}}{\delta}\right\rangle \tag{B.35b}$$

The error bounds (2.50) were obtained using no smoothness assumption on f and h; they also hold when full gradients of f are replaced with elements in the subgraident of f. The bounds was also obtained without *using the arbitrary update $y_{k+1} = \mathcal{G}(x)$. In particular, accelerated methods are obtained by picking a map $\mathcal{G}$ that results in a better bound on the error than the straight forward discretization $y_{k+1} = x_{k+1}$. We immediately see that any algorithm for which the map $\mathcal{G}$ satisfies the progress condition $f(y_{k+1}) - f(x_{k+1}) \propto -\|\nabla f(x_{k+1})\|^{\frac{p}{p-1}}$ or $\langle \nabla f(y_{k+1}), y_{k+1} - x_{k+1}\rangle \propto -\|\nabla f(y_{k+1})\|^{\frac{p}{p-1}}$ will obtain a $O(1/\epsilon\sigma k^p)$ convergence rate. We now present short descriptions of the main results for the aforementioned five papers.*

B.6.1 Proof of Proposition B.6.1

We show the initial bounds (B.35a) and (B.35b). We begin with algorithm (2.46):

$$\begin{aligned}
\frac{E_{k+1} - E_k}{\delta} &= -\left\langle \frac{\nabla h(z_{k+1}) - \nabla h(z_k)}{\delta}, x^* - z_{k+1} \right\rangle - \frac{1}{\delta} D_h(z_{k+1}, z_k) + A_{k+1} \frac{f(y_{k+1}) - f(x_{k+1})}{\delta} \\
&\quad + \frac{A_{k+1} - A_k}{\delta} (f(x_{k+1}) - f(x^*)) + A_k \frac{f(x_{k+1}) - f(y_k)}{\delta} \\
&\overset{(2.46b)}{=} \alpha_k \langle \nabla f(x_{k+1}), x^* - z_{k+1}\rangle - \frac{1}{\delta} D_h(z_{k+1}, z_k) + A_{k+1} \frac{f(y_{k+1}) - f(x_{k+1})}{\delta} \\
&\leq \alpha_k \langle \nabla f(x_{k+1}), x^* - z_k\rangle + \alpha_k \langle \nabla f(x_{k+1}), z_k - z_{k+1}\rangle - \frac{\sigma}{\delta p} \|z_{k+1} - z_k\|^p \\
&\quad + A_{k+1} \frac{f(y_{k+1}) - f(x_{k+1})}{\delta} + \frac{A_{k+1} - A_k}{\delta} (f(x_{k+1}) - f(x^*)) + A_k \frac{f(x_{k+1}) - f(y_k)}{\delta} \\
&\leq \alpha_k \langle \nabla f(x_{k+1}), x^* - z_k\rangle A_k \frac{f(x_{k+1}) - f(y_k)}{\delta} + \frac{A_{k+1} - A_k}{\delta} (f(x_{k+1}) - f(x^*)) \\
&\quad + \frac{p-1}{p} \sigma^{-\frac{1}{p-1}} \delta^{-1} (A_{k+1} - A_k)^{\frac{p}{p-1}} \|\nabla f(x_{k+1})\|^{\frac{p}{p-1}} + A_{k+1} \frac{f(y_{k+1}) - f(x_{k+1})}{\delta}
\end{aligned}$$

The first inequality follows from the σ-uniform convexity of h with respect to the p-th power of the norm and the last inequality follows from Young's inequality (A.25). If we continue with our argument, and plug in the identity (B.35a), it simply remains to use our second update (2.46a):

$$\begin{aligned}
\frac{E_{k+1} - E_k}{\delta} &\leq \alpha_k \langle \nabla f(x_{k+1}), x^* - z_k\rangle + A_k \frac{f(x_{k+1}) - f(y_k)}{\delta} + \frac{A_{k+1} - A_k}{\delta} (f(x_{k+1}) - f(x^*)) \\
&\quad + \frac{p-1}{p} \sigma^{-\frac{1}{p-1}} \delta^{-1} (A_{k+1} - A_k)^{\frac{p}{p-1}} \|\nabla f(x_{k+1})\|^{\frac{p}{p-1}} + A_{k+1} \frac{f(y_{k+1}) - f(x_{k+1})}{\delta} \\
&\leq \alpha_k \langle \nabla f(x_{k+1}), x^* - y_k\rangle + A_{k+1} \left\langle \nabla f(x_{k+1}), \frac{y_k - x_{k+1}}{\delta} \right\rangle + A_k \frac{f(x_{k+1}) - f(y_k)}{\delta} \\
&\quad + \frac{A_{k+1} - A_k}{\delta} (f(x_{k+1}) - f(x^*)) + \varepsilon_{k+1} \\
&= -\frac{A_{k+1} - A_k}{\delta} D_f(x^*, x_{k+1}) - A_{k+1}/\delta D_f(x_{k+1}, y_k) + \varepsilon_{k+1}
\end{aligned}$$

From here, we can conclude $\frac{E_{k+1}-E_k}{\delta} \leq \varepsilon_k$ *using the convexity of* f.

We now show the bound (B.35b) *for algorithm* (2.47)

$$\begin{aligned}
\frac{E_{k+1}-E_k}{\delta} &= \frac{D_h\left(x^*, z_{k+1}\right) - D_h\left(x^*, z_k\right)}{\delta} + A_{k+1}\left(f\left(y_{k+1}\right) - f(x)\right) - A_k\left(f(y_k) - f(x^*)\right) \\
&\overset{(2.47c)}{=} \alpha_k \left\langle \nabla f(y_{k+1}), x^* - z_{k+1}\right\rangle - \frac{1}{\delta} D_h\left(z_{k+1}, z_k\right) + \frac{A_{k+1}-A_k}{\delta}\left(f\left(y_{k+1}\right) - f\left(x^*\right)\right) \\
&\quad + A_k \frac{f\left(y_{k+1}\right) - f\left(y_k\right)}{\delta} \\
&\leq \alpha_k \left\langle \nabla f\left(y_{k+1}\right), x^* - z_k\right\rangle + \alpha_k \left\langle \nabla f\left(y_{k+1}\right), z_k - z_{k+1}\right\rangle - \frac{\sigma}{\delta p}\|z_{k+1} - z_k\|^p \\
&\quad + \frac{A_{k+1}-A_k}{\delta}\left(f\left(y_{k+1}\right) - f\left(x^*\right)\right) + A_k \frac{f\left(y_{k+1}\right) - f\left(y_k\right)}{\delta} \\
&\leq \alpha_k \left\langle \nabla f(y_{k+1}), x^* - z_k\right\rangle + A_k \frac{f\left(y_{k+1}\right) - f\left(y_k\right)}{\delta} + \frac{A_{k+1}-A_k}{\delta}\left(f\left(y_{k+1}\right) - f(x^*)\right) \\
&\quad - A_{k+1}\left\langle \nabla f(y_{k+1}), \frac{y_{k+1} - x_{k+1}}{\delta}\right\rangle + \varepsilon_{k+1}
\end{aligned}$$

The first inequality follows from the uniform convexity of h *and the second uses Young's inequality* (A.25) *and definition* (B.35b). *Using the second update* (2.47a), *we obtain our initial error bound:*

$$\begin{aligned}
\frac{E_{k+1}-E_k}{\delta} &\leq \alpha_k \left\langle \nabla f\left(y_{k+1}\right), x^* - y_k\right\rangle + A_k \frac{f\left(y_{k+1}\right) - f\left(y_k\right)}{\delta} + \frac{A_{k+1}-A_k}{\delta} f\left(y_{k+1}\right) - f\left(x^*\right) \\
&\quad + A_{k+1}\left\langle \nabla f\left(y_{k+1}\right), \frac{y_k - x_{k+1}}{\delta}\right\rangle - A_{k+1}\left\langle \nabla f\left(y_{k+1}\right), \frac{y_{k+1} - x_{k+1}}{\delta}\right\rangle + \varepsilon_{k+1} \\
&= \alpha_k D_f(x^*, y_{k+1}) - A_k/\delta D_f(y_{k+1}, y_k) + \varepsilon_{k+1}
\end{aligned}$$

Accelerated universal methods [40, 6, 46, 21] *The term "universal methods" refers to the algorithms designed for the class of functions with* (ϵ, ν)*-Holder-continuous higher-order gradients* ($2 \leq p \in \mathbb{N}$, $\nu \in (0, 1]$, $\epsilon > 0$),

$$\|\nabla^{p-1} f(x) - \nabla^{p-1} f(y)\| \leq \frac{1}{\epsilon}\|x - y\|^{\nu}. \tag{B.36}$$

Typically, practitioners care about the setting where we have Holder continuous gradients ($p = 2$) *or Holder-continuous Hessians* ($p = 3$), *since methods which use higher-order information are often too computationally expensive. In the case* $p \geq 3$, *the gradient update*

$$\mathcal{G}_{\epsilon,p,\nu,N}(x) = \arg\min_{y \in \mathcal{X}} \left\{ f_{p-1}(x; y) + \frac{N}{\epsilon \tilde{p}}\|x - y\|^{\tilde{p}} \right\}, \quad \tilde{p} = p - 1 + \nu,\ N > 1 \tag{B.37}$$

can be used to simplify the error (2.50b) *obtained by algorithm* (2.47). *Notice, the gradient update is regularized by the smoothness parameter* $\tilde{p}$. *We summarize this result in the following proposition.*

Lemma B.6.2. *Assume f has Holder continuous higher-order gradients. Using the map $y_{k+1} = \mathcal{G}_{\epsilon,p,\nu,N}(x_{k+1})$ defined by (B.37) in update (2.47b), results in the following progress condition,*

$$\langle \nabla f(y_{k+1}), y_{k+1} - x_{k+1} \rangle \leq -\frac{(N^2-1)^{\frac{\tilde{p}-1}{2\tilde{p}-2}}}{2N} \epsilon^{\frac{1}{\tilde{p}-1}} \|\nabla f(y_{k+1})\|^{\frac{\tilde{p}}{\tilde{p}-1}}, \tag{B.38}$$

where $\tilde{p} = p - 1 + \nu$ and $p \geq 3$.

Lemma B.6.2 demonstrates that if the Taylor approximation is regularized according to the smoothness of the function, the progress condition scales as a function of the smoothness in a particularly nice way. Using this inequality, we can simplify the error (2.50b) in algorithm (2.47) to the following,

$$\begin{aligned}\varepsilon_{k+1} = &\frac{\tilde{p}-1}{\tilde{p}} \sigma^{-\frac{1}{\tilde{p}-1}} \frac{(A_{k+1}-A_k)^{\frac{\tilde{p}}{\tilde{p}-1}}}{\delta} \|\nabla f(y_{k+1})\|^{\frac{\tilde{p}}{\tilde{p}-1}} \\ &- \frac{A_{k+1}}{\delta} \frac{(N^2-1)^{\frac{\tilde{p}-1}{2\tilde{p}-2}}}{2N} \epsilon^{\frac{1}{\tilde{p}-1}} \|\nabla f(y_{k+1})\|^{\frac{\tilde{p}}{\tilde{p}-1}},\end{aligned}$$

where we have assumed that the geometry scales nicely with the smoothness condition: $D_h(x,y) \geq \frac{\sigma}{\tilde{p}}\|x-y\|^{\tilde{p}}$. This requires the condition $p \geq 3$. To ensure a non-positive error we choose a sequence which satisfies the bound,

$$\frac{(A_{k+1}-A_k)^{\frac{\tilde{p}}{\tilde{p}-1}}}{A_{k+1}} \leq (\epsilon\sigma)^{\frac{1}{\tilde{p}-1}} \frac{\tilde{p}}{\tilde{p}-1} \frac{(N^2-1)^{\frac{\tilde{p}-1}{2\tilde{p}-2}}}{2N} := C_{\epsilon,\sigma,\tilde{p},N}.$$

This bound is maximized by polynomials in k of degree $\tilde{p}$ with leading coefficient proportional to $C^{\tilde{p}-1}_{\epsilon,\sigma,\tilde{p},N}$; this results in the convergence rate bound $f(y_k) - f(x^*) \leq O(1/\epsilon\sigma k^{\tilde{p}}) = O(1/\epsilon\sigma k^{p-1+\nu})$. We can compare this convergence rate, to that obtained by using just gradient map $y_{k+1} = \mathcal{G}_{\epsilon,p,\tilde{p},N}(y_k)$; this algorithm obtains a slower $f(y_k) - f(x^*) \leq O(1/\epsilon\sigma k^{\tilde{p}-1}) = O(1/\epsilon\sigma k^{p-2+\nu})$ convergence rate under the same smoothness assumptions. This result unifies and extends the analyses of the accelerated (universal) cubic regularized Newton's method [40, 21] and accelerated higher-order methods [6]. Wibisono et. al. [76] show that $\|x_k - y_k\| = O(\epsilon^{1/\tilde{p}})$ and $\varepsilon_k = O(\epsilon^{1/\tilde{p}})$ so that as $\epsilon^{1/\tilde{p}} \to 0$ we recover the dynamics (2.41) and the statement $\dot{\mathcal{E}}_t \leq 0$ for Lyapunov function (2.38).

We end by mentioning that in the special case $p = 2$, Nesterov [46] showed that a slightly modified gradient map,

$$\mathcal{G}_{\tilde{\epsilon}}(x) = x - \tilde{\epsilon}\, \nabla f(x), \tag{B.39}$$

has the following property when applied to functions with Holder continuous gradients.

Lemma B.6.3. *([46, Lemma 1]) Assume f has (ϵ, ν)-Holder continuous gradients, where $\nu \in (0, 1]$. Then for $1/\tilde{\epsilon} \geq (1/2\tilde{\delta})^{\frac{1-\nu}{1+\nu}}(1/\epsilon)^{\frac{2}{1+\nu}}$ the following bound,*

$$f(y_{k+1}) - f(x_{k+1}) \leq -\frac{\tilde{\epsilon}}{2}\|\nabla f(x_{k+1})\|^2 + \tilde{\delta},$$

holds for $y_{k+1} = \mathcal{G}_{\tilde{\epsilon}}(x_{k+1})$ given by (B.39).

That is, if we take a gradient descent step with increased regularization and assume h is σ-strongly convex, the error for algorithm (2.46) when f is (ϵ, ν)-Holder continuous can be written as,

$$\varepsilon_{k+1} = \frac{(A_{k+1} - A_k)^2}{2\sigma\delta}\|\nabla f(x_{k+1})\|^2 - \frac{\tilde{\epsilon}A_{k+1}}{2\delta}\|\nabla f(x_{k+1})\|^2 + \tilde{\delta}. \tag{B.40}$$

This allows us to conclude $O(1/\tilde{\epsilon}\sigma k^2)$ convergence rate of the function to within $\tilde{\delta}$, which is controlled by the amount of regularization $\tilde{\epsilon}$ we apply in the gradient update.

B.6.2 Proof of Lemma B.6.2

A similar progress bound was proved in Wibisono, Wilson and Jordan [76, Lem 3.2]. Note, $y = \mathcal{G}(x)$ satisfies the optimality condition

$$\sum_{i=1}^{p-1} \frac{1}{(i-1)!}\nabla^i f(x)\,(y-x)^{i-1} + \frac{N}{\epsilon}\|y-x\|^{\tilde{p}-2}\,(y-x) = 0. \tag{B.41}$$

Furthermore, since $\nabla^{p-1} f$ is Holder-continuous (B.36), we have the following error bound on the $(p-2)$-nd order Taylor expansion of ∇f,

$$\begin{aligned}\left\|\nabla f(y) - \sum_{i=0}^{p-1} \frac{1}{(i-1)!}\nabla^i f(x)(y-x)^{i-1}\right\| &= \left\|\int_0^1 [\nabla^{p-1} f(ty + (1-t)x) - \nabla^{p-1} f(x)](y-x)^{p-2} dt\right\| \\ &\leq \frac{1}{\epsilon}\|y-x\|^{p-2+\nu}\int_0^1 t^\nu = \frac{1}{\epsilon}\|y-x\|^{\tilde{p}-1}\end{aligned} \tag{B.42}$$

Substituting (B.41) to (B.42) and writing $r = \|y-x\|$, we obtain

$$\left\|\nabla f(y) + \frac{Nr^{\tilde{p}-2}}{\epsilon}(y-x)\right\|_* \leq \frac{r^{\tilde{p}-1}}{\epsilon}. \tag{B.43}$$

Now the argument proceeds as in [76]. Squaring both sides, expanding, and rearranging the terms, we get the inequality

$$\langle \nabla f(y), x - y\rangle \geq \frac{\epsilon}{2Nr^{\tilde{p}-2}}\|\nabla f(y)\|_*^2 + \frac{(N^2-1)r^{\tilde{p}}}{2N\epsilon}. \tag{B.44}$$

Note that if $\tilde{p} = 2$, then the first term in (B.44) already implies the desired bound (B.38). Now assume $\tilde{p} \geq 3$. The right-hand side of (B.44) is of the form $A/r^{\tilde{p}-2} + Br^{\tilde{p}}$, which is a convex function of $r > 0$ and minimized by $r^* = \left\{\frac{(\tilde{p}-2)}{\tilde{p}}\frac{A}{B}\right\}^{\frac{1}{2\tilde{p}-2}}$, yielding a minimum value of

$$\frac{A}{(r^*)^{\tilde{p}-2}} + B(r^*)^p \;=\; A^{\frac{\tilde{p}}{2\tilde{p}-2}} B^{\frac{\tilde{p}-2}{2\tilde{p}-2}} \left[\left(\frac{\tilde{p}}{\tilde{p}-2}\right)^{\frac{\tilde{p}-2}{2\tilde{p}-2}} + \left(\frac{\tilde{p}-2}{\tilde{p}}\right)^{\frac{\tilde{p}}{2\tilde{p}-2}}\right] \;\geq\; A^{\frac{\tilde{p}}{2\tilde{p}-2}} B^{\frac{\tilde{p}-2}{2\tilde{p}-2}}.$$

Substituting the values $A = \frac{\epsilon}{2N}\|\nabla f(y)\|_*^2$ and $B = \frac{1}{2N\epsilon}(N^2-1)$ from (B.44), we obtain

$$\langle \nabla f(y), x-y\rangle \;\geq\; \left(\frac{\epsilon}{2N}\|\nabla f(y)\|_*^2\right)^{\frac{\tilde{p}}{2\tilde{p}-2}} \left(\frac{1}{2N\epsilon}(N^2-1)\right)^{\frac{\tilde{p}-2}{2\tilde{p}-2}} = \frac{(N^2-1)^{\frac{\tilde{p}-2}{2\tilde{p}-2}}}{2N}\epsilon^{\frac{1}{\tilde{p}-1}}\|\nabla f(y)\|_*^{\frac{\tilde{p}}{\tilde{p}-1}},$$

which proves the progress bound (B.38).

B.6.3 Proof of Proposition 2.2.4

We show the initial error bound (2.64). We check,

$$\begin{aligned}
\frac{E_{k+1}-E_k}{\delta} &= \frac{A_{k+1}-A_k}{\delta}\left(f\left(y_k\right) - f\left(x^*\right) - \mu D_h\left(x^*, z_k\right)\right) + A_{k+1}\frac{f\left(y_{k+1}\right) - f\left(y_k\right)}{\delta} \\
&\quad - A_{k+1}\mu\left\langle \frac{\nabla h\left(z_{k+1}\right) - \nabla h\left(z_k\right)}{\delta}, x^* - z_k\right\rangle + \varepsilon_k^1 \\
&\leq \frac{A_{k+1}-A_k}{\delta}\left(f\left(y_k\right) - f\left(x^*\right) - \mu D_h\left(x^*, z_k\right)\right) + A_{k+1}\left\langle \nabla f(x_k), \frac{x_k - y_k}{\delta}\right\rangle \\
&\quad - \frac{\mu}{\delta}A_{k+1}D_h\left(x_k, y_k\right) - A_{k+1}\mu\left\langle \frac{\nabla h\left(z_{k+1}\right) - \nabla h\left(z_k\right)}{\delta}, x^* - z_k\right\rangle + \varepsilon_k^2 \\
&\overset{(2.62b)}{=} \frac{A_{k+1}-A_k}{\delta}\left(f(y_k) - f(x^*) - \mu D_h\left(x^*, z_k\right)\right) + A_{k+1}\left\langle \nabla f(x_k), \frac{x_k - y_k}{\delta}\right\rangle \\
&\quad - \frac{\mu}{\delta}A_{k+1}D_h\left(x_k, y_k\right) + A_{k+1}\tau_k\left\langle \nabla f(x_k), x^* - x_k\right\rangle + A_{k+1}\tau_k\left\langle \nabla f\left(x_k\right), x_k - z_k\right\rangle \\
&\quad - \mu A_{k+1}\tau_k\left\langle \nabla h\left(x_k\right) - \nabla h\left(z_k\right), x^* - z_k\right\rangle + \varepsilon_k^2 \\
&= \frac{A_{k+1}-A_k}{\delta}\left(f(y_k) - f(x^*) - \mu D_h\left(x^*, z_k\right)\right) - \frac{\mu}{\delta}A_{k+1}D_h\left(x_k, y_k\right) \\
&\quad + A_{k+1}\tau_k\left\langle \nabla f(x_k), x^* - x_k\right\rangle - \mu A_{k+1}\tau_k\left\langle \nabla h\left(x_k\right) - \nabla h\left(z_k\right), x^* - z_k\right\rangle + \varepsilon_k^2 \\
&\leq \frac{A_{k+1}-A_k}{\delta}\left(f\left(y_k\right) - f\left(x^*\right) - \mu D_h\left(x^*, z_k\right)\right) - A_{k+1}\tau_k\left(f\left(x_k\right) - f\left(x^*\right) + \mu D_h\left(x^*, x_k\right)\right) \\
&\quad - A_{k+1}\frac{\sigma\mu}{2\delta}\|x_k - y_k\|^2 - A_{k+1}\tau_k\mu\left\langle \nabla h\left(x_k\right) - \nabla h\left(z_k\right), x^* - z_k\right\rangle + \varepsilon_k^2
\end{aligned}$$

Here, $\varepsilon_k^1 = A_{k+1}\mu\left(\left(\langle \nabla h\left(z_{k+1}\right) - \nabla h\left(z_k\right)\right)/\delta, z_k - z_{k+1}\rangle + \frac{1}{\delta}D_h\left(z_{k+1}, z_k\right)\right) \leq A_{k+1}\frac{\sigma\mu}{2\delta}\|\nabla h\left(z_{k+1}\right) - \nabla h(z_k)\|^2$, *where the upper bound follows from the* σ*-strong convexity of* h *and Young's inequality.* $\varepsilon_k^2 = A_{k+1}\frac{f(y_{k+1})-f(x_k)}{\delta} + A_{k+1}\frac{\sigma\mu}{2\delta}\|\nabla h\left(z_{k+1}\right) - \nabla h\left(z_k\right)\|^2$. *The first inequality uses the* μ*-strong convexity of* f *with respect to* h. *The second inequality uses the strong convexity of* f *and* σ*-strong convexity of* h. *We continue by using the Bregman three point identity (A.27)*

$$\begin{aligned}\frac{E_{k+1}-E_k}{\delta} &= \frac{A_{k+1}-A_k}{\delta}\left(f\left(y_k\right) - f\left(x_k\right) + \mu D_h\left(x_k, z_k\right)\right) - A_{k+1}\frac{\sigma\mu}{2\delta}\|x_k - y_k\|^2 + \varepsilon_k^2 \\ &\leq \frac{A_{k+1}-A_k}{\delta}\left(\langle \nabla f\left(x_k\right), y_k - x_k\rangle + (1/\epsilon)D_h\left(y_k, x_k\right) - \mu D_h\left(x_k, z_k\right)\right) \\ &\quad - A_{k+1}\frac{\sigma\mu}{2\delta}\|x_k - y_k\|^2 + \varepsilon_k^2\end{aligned}$$

The last line follows from using the $(1/\epsilon)$*-smoothness of* f. *Now we turn to the case where* $h = \frac{1}{2}\|x\|^2$ *(so* $\sigma = 1$*)*

$$\begin{aligned}\frac{E_{k+1}-E_k}{\delta} &\leq \frac{A_{k+1}-A_k}{\delta}\left(\langle \nabla f\left(x_k\right), y_k - x_k\rangle + \frac{1}{2\epsilon}\|y_k - x_k\|^2 - \frac{\mu}{2}\|x_k - z_k\|^2\right) \\ &\quad - \delta A_{k+1}\frac{\mu}{2}\left\|\frac{x_k - y_k}{\delta}\right\|^2 + \varepsilon_k^2 + A_{k+1}\frac{f\left(y_{k+1}\right) - f\left(x_k\right)}{\delta} + \delta A_{k+1}\frac{\mu}{2}\left\|\frac{z_{k+1} - z_k}{\delta}\right\|^2 \\ &= \frac{A_{k+1}-A_k}{\delta}\left(\langle \nabla f\left(x_k\right), y_k - x_k\rangle + \frac{1}{2\epsilon}\|y_k - x_k\|^2 - \frac{\mu}{2\tau_k^2\delta^2}\|x_k - y_k\|^2\right) \\ &\quad - \delta A_{k+1}\tau_k^2\frac{\mu}{2}\|z_k - x_k\|^2 + \varepsilon_k^2 + A_{k+1}\frac{f\left(y_k + 1\right) - f\left(x_k\right)}{\delta} \\ &\quad + \delta A_{k+1}\frac{\mu}{2}\|\tau_k\left(x_k - z_k - (1/\mu)\nabla f\left(x_k\right)\right)\|^2 \\ &= -\delta A_{k+1}\left(\frac{A_{k+1}\mu}{2\alpha_k\delta} - \frac{\alpha_k\delta}{2A_{k+1}\epsilon}\right)\left\|\frac{y_k - x_k}{\delta}\right\|^2 \\ &\quad + A_{k+1}\left(\frac{f\left(y_{k+1}\right) - f\left(x_k\right)}{\delta} + \delta\tau_k^2\frac{1}{2\mu}\|\nabla f\left(x_k\right)\|^2\right)\end{aligned}$$

B.6.4 Proof of Theorem 2.2.6

We follow the framework of Su, Boyd and Candes [70, pg. 36]. It suffices to established that our Lyapunov function is monotonically decreasing. Although $\mathcal{E}_t$ *may not be differentiable, we can study* $\mathcal{E}(t+\Delta t) - \mathcal{E}(t))/\Delta t$ *for small* $\Delta t > 0$. *For the first term in (2.65), note that*

$$\begin{aligned}(t+\Delta t)^p(f(X_{t+\Delta t}) - f(x)) - t^p(f(X_t) - f(x)) &= t^p(f(X_{t+\Delta t}) - f(X_t)) \\ &\quad + pt^{p-1}(f(X_{t+\Delta t}) - f(x))\Delta t + o(\Delta t) \\ &= t^p\langle G_f(X_t, \dot{X}_t), \dot{X}_t\rangle\Delta t \\ &\quad + pt^{p-1}(f(X_{t+\Delta t}) - f(x))\Delta t + o(\Delta t)\end{aligned}$$

where the second line follows since we assume f is locally Lipschitz, the $o(\Delta t)$ does not affect the function in the limit:

$$\begin{aligned} f(X_{t+\Delta t}) = f(X + \Delta t \dot{X}_t + o(\Delta t)) &= f(X + \Delta t \dot{X}_t) + o(\Delta t) \\ &= f(X_t) + \langle G_f(X_t, \dot{X}_t), \dot{X}_t \rangle \Delta t + o(\Delta t) \end{aligned} \tag{B.45}$$

The second term $D_h(x, X_t + \frac{t}{p}\dot{X}_t)$ is differentiable, with derivative $-\left\langle \frac{d}{dt}\nabla h(Z_t), x - Z_t \right\rangle$. Hence,

$$\begin{aligned} &D_h\Big(x, X_{t+\Delta t} + \frac{t+\Delta t}{p}\dot{X}_{t+\Delta t}\Big) - D_h\Big(x, X_t + \frac{t}{p}\dot{X}_t\Big) \\ &= -\left\langle \frac{d}{dt}\nabla h(Z_t), x - Z_t \right\rangle \Delta t + o(\Delta t) \\ &= pt^{p-1}\langle G_\varphi(X_t, \dot{X}_t), x - Z_t \rangle \Delta t + o(\Delta t) \\ &= pt^{p-1}\langle G_\varphi(X_t, \dot{X}_t), x - X_t \rangle \Delta t + t^p \langle G_\varphi(X_t, \dot{X}_t), \dot{X}_t \rangle \Delta t + o(\Delta t) \\ &\leq -pt^{p-1}(f(X_t) - f(x))\Delta t + t^p \langle G_\varphi(X_t, \dot{X}_t), \dot{X}_t \rangle \Delta t + o(\Delta t) \\ &= -pt^{p-1}(f(X_t) - f(x))\Delta t + t^p \langle G_f(X_t, \dot{X}_t), \dot{X}_t \rangle \Delta t \end{aligned}$$

The inequality follows from the convexity of f.Combining everything we have shown

$$\limsup_{\Delta t \to 0^+} \frac{\mathcal{E}_{t+\Delta t} - \mathcal{E}_t}{\Delta t} \leq 0,$$

which along with the continuity of $\mathcal{E}_t$, ensures $\mathcal{E}_t$ is a non-increasing of time. We can make a similar argument for dynamic (2.66). Notice the first term in the approximation $\mathcal{E}(t+\Delta t) - \mathcal{E}(t))/\Delta t$, is the same as in the previous setting. Therefore we calculate the second term,

$$\begin{aligned} &(t^p + \Delta t)\left(D_h\Big(x, X_{t+\Delta t} + \frac{t+\Delta t}{p}\dot{X}_{t+\Delta t}\Big) - D_h\Big(x, X_t + \frac{t}{p}\dot{X}_t\Big)\right) - pt^{p-1} D_h\Big(x, X_t + \frac{t}{p}\dot{X}_t\Big)\Delta t \\ &= (t^p + \Delta t)\left(-\left\langle \frac{d}{dt}\nabla h(Z_t), x - Z_t \right\rangle \Delta t + o(\Delta t)\right) - pt^{p-1} D_h\Big(x, X_t + \frac{t}{p}\dot{X}_t\Big)\Delta t \\ &\overset{(2.66)}{=} pt^{p-1}\left(-\langle \nabla h(X_t) - \nabla h(Z_t), x - Z_t \rangle \Delta t + \left\langle G(X_t, \dot{X}_t), x - Z_t \right\rangle \Delta t - D_h(x, Z_t)\Delta t + \Delta t\right) \\ &\overset{(A.27)}{=} pt^{p-1}\left(D_h(x^*, X_t)\Delta t + \left\langle G(X_t, \dot{X}_t), x - Z_t \right\rangle \Delta t\right) \\ &\leq -pt^{p-1}(f(X_t) - f(x))\Delta t + t^p \langle G_f(X_t, \dot{X}_t), \dot{X}_t \rangle \Delta t \end{aligned}$$

where the last line follows from convexity. Combining everything we have shown

$$\limsup_{\Delta t \to 0^+} \frac{\mathcal{E}_{t+\Delta t} - \mathcal{E}_t}{\Delta t} \leq 0,$$

which along with the continuity of $\mathcal{E}_t$, ensures $\mathcal{E}_t$ is a non-increasing of time.

B.7 Estimate Sequences

In this section we formalize the connection between estimate sequences and Lyapunov functions.

B.7.1 The Quasi-Montone Subgradient Method

The discrete-time estimate sequence (2.72) *for quasi-monotone subgradient method can be written:*

$$\begin{aligned}\phi_{k+1}(x) - A_{k+1}^{-1}\tilde{\varepsilon}_{k+1} &:= f(x_{k+1}) + A_{k+1}^{-1}D_h(x, z_{k+1}) - A_{k+1}^{-1}\tilde{\varepsilon}_{k+1} \\ &\overset{(2.72)}{=} (1-\tau_k)\left(\phi_k(x) - A_k^{-1}\tilde{\varepsilon}_k\right) + \tau_k f_k(x) \\ &= \left(1 - \frac{\alpha_k}{A_{k+1}}\right)\left(f(x_k) + \frac{1}{A_k}D_h(x, z_k) - \frac{\tilde{\varepsilon}_k}{A_k}\right) + \frac{\alpha_k}{A_{k+1}}f_k(x).\end{aligned}$$

Multiplying through by A_{k+1}*, we have*

$$\begin{aligned}A_{k+1}f(x_{k+1}) + D_h(x, z_{k+1}) - \tilde{\varepsilon}_{k+1} &= (A_{k+1} - \alpha_k)(f(x_k) + A_k^{-1}D_h(x, z_k) - A_k^{-1}\tilde{\varepsilon}_k) \\ &\quad - (A_{k+1} - \alpha_k)A_k^{-1}\tilde{\varepsilon}_k + \alpha_k f_k(x) \\ &= A_k\left(f(x_k) + A_k^{-1}D_h(x, z_k) - A_k^{-1}\tilde{\varepsilon}_k\right) + \alpha_k f_k(x) \\ &\overset{(2.71)}{\leq} A_k f(x_k) + D_h(x, z_k) - \tilde{\varepsilon}_k + \alpha_k f(x).\end{aligned}$$

Rearranging, we obtain our Lyapunov argument $E_{k+1} \leq E_k + \varepsilon_{k+1}$ *for* (2.48)*:*

$$A_{k+1}(f(x_{k+1}) - f(x)) + D_h(x, z_{k+1}) \leq A_k(f(x_k) - f(x)) + D_h(x, z_k) + \varepsilon_{k+1}.$$

Going the other direction, from our Lyapunov analysis we can derive the following bound:

$$E_k \leq E_0 + \tilde{\varepsilon}_k \tag{B.46}$$

$$\begin{aligned}A_k(f(x_k) - f(x)) + D_h(x, z_k) &\leq A_0(f(x_0) - f(x)) + D_h(x, z_0) + \tilde{\varepsilon}_k \\ A_k\left(f(x_k) - \frac{1}{A_k}D_h(x, z_k)\right) &\leq (A_k - A_0)f(x) + A_0\left(f(x_0) + \frac{1}{A_0}D_h(x^*, z_0)\right) + \tilde{\varepsilon}_k \\ A_k\phi_k(x) &\leq (A_k - A_0)f(x) + A_0\phi_0(x) + \tilde{\varepsilon}_k.\end{aligned} \tag{B.47}$$

Rearranging, we obtain our estimate sequence (2.69) *(*$A_0 = 1$*) with an additional error term:*

$$\phi_k(x) \leq \left(1 - \frac{A_0}{A_k}\right)f(x) + \frac{A_0}{A_k}\phi_0(x) + \frac{\tilde{\varepsilon}_k}{A_k} = \left(1 - \frac{1}{A_k}\right)f(x) + \frac{1}{A_k}\phi_0(x) + \frac{\tilde{\varepsilon}_k}{A_k}. \tag{B.48a}$$

B.7.2 Frank-Wolfe

The discrete-time estimate sequence (2.72) for conditional gradient method can be written:

$$\phi_{k+1}(x) - \frac{\tilde{\varepsilon}_{k+1}}{A_{k+1}} := f(x_{k+1}) - \frac{\tilde{\varepsilon}_{k+1}}{A_{k+1}} \overset{(2.72)}{=} (1-\tau_k)\left(\phi_k(x) - \frac{\tilde{\varepsilon}_k}{A_k}\right) + \tau_k f_k(x)$$
$$\overset{\text{Table 2.8}}{=} \left(1 - \frac{\alpha_k}{A_{k+1}}\right)\left(f(x_k) - \frac{\tilde{\varepsilon}_k}{A_k}\right) + \frac{\alpha_k}{A_{k+1}} f_k(x).$$

Multiplying through by A_{k+1}, we have

$$\begin{aligned} A_{k+1}\left(f(x_{k+1}) - \frac{\tilde{\varepsilon}_{k+1}}{A_{k+1}}\right) &= (A_{k+1} - (A_{k+1} - A_k))\left(f(x_k) - \frac{\tilde{\varepsilon}_k}{A_k}\right) + \alpha_k f_k(x) \\ &= A_k\left(f(x_k) - A_k^{-1}\tilde{\varepsilon}_k\right) + (A_{k+1} - A_k) f_k(x) \\ &\overset{(2.71)}{\leq} A_k f(x_k) - \tilde{\varepsilon}_k + (A_{k+1} - A_k) f(x). \end{aligned}$$

Rearranging, we obtain our Lyapunov argument $E_{k+1} - E_k \leq \varepsilon_{k+1}$ for (2.35) :

$$A_{k+1}(f(x_{k+1}) - f(x)) \leq A_k(f(x_k) - f(x)) + \varepsilon_{k+1}.$$

Going the other direction, from our Lyapunov analysis we can derive the following bound:

$$\begin{aligned} E_k &\leq E_0 + \tilde{\varepsilon}_k \\ A_k f(x_k) &\leq (A_k - A_0) f(x) + A_0 f(x_0) + \tilde{\varepsilon}_k \\ A_k \phi_k(x) &\leq (A_k - A_0) f(x) + A_0 \phi_0(x) + \tilde{\varepsilon}_k \end{aligned}$$

Rearranging, we obtain our estimate sequence (2.69) ($A_0 = 1$) with an additional error term:

$$\phi_k(x) \leq \left(1 - \frac{A_0}{A_k}\right) f(x) + \frac{A_0}{A_k}\phi_0(x) + \frac{\tilde{\varepsilon}_k}{A_k} = \left(1 - \frac{1}{A_k}\right) f(x) + \frac{1}{A_k}\phi_0(x) + \frac{\tilde{\varepsilon}_k}{A_k}.$$

Since the Lyapunov function property allows us to write

$$e^{\beta_t} f(X_t) \leq (e^{\beta_t} - e^{\beta_0}) f(x) + e^{\beta_0} f(X_0),$$

we can extract $\{f(X_t), e^{\beta_t}\}$ as the continuous-time estimate sequence for Frank-Wolfe.

B.7.3 Accelerated Gradient Descent (Strong Convexity)

The discrete-time estimate sequence (2.72) for accelerated gradient descent can be written:

$$\phi_{k+1}(x) := f(x_{k+1}) + \frac{\mu}{2}\|x - z_{k+1}\|^2 \overset{(2.72)}{=} (1-\tau_k)\phi_k(x) + \tau_k f_k(x) \overset{(2.71)}{\leq} (1-\tau_k)\phi_k(x) + \tau_k f(x).$$

Therefore, we obtain the inequality $\tilde{E}_{k+1} - \tilde{E}_k \leq -\tau_k \tilde{E}_k$ *for our Lyapunov function* $\tilde{E}_k = f(x_k) - f(x^a st) + \frac{\mu}{2}\|x^* - x_k\|^2$ *by simply writing* $\phi_{k+1}(x) - f(x) + f(x) - \phi_k(x) \leq -\tau_k(\phi_k(x) - f(x))$*:*

$$f(x_{k+1}) - f(x) + \frac{\mu}{2}\|x - z_{k+1}\|^2 - \left(f(x_k) - f(x) + \frac{\mu}{2}\|x - z_{k+1}\|^2\right) \overset{\text{Table 2.8}}{\leq} -\tau_k \left(f(x_k) - f(x) + \frac{\mu}{2}\|x - z_{k+1}\|^2\right).$$

Going the other direction, we have,

$$\begin{aligned} E_{k+1} - E_k &\leq -\tau_k E_k \\ \phi_{k+1} &\leq (1 - \tau_k)\phi_k(x) + \tau_k f(x) \\ A_{k+1}\phi_{k+1} &\leq A_k \phi_k + (A_{k+1} - A_k) f(x). \end{aligned}$$

Summing over the right-hand side, we obtain the estimate sequence (2.69)*:*

$$\phi_{k+1} \leq \left(1 - \frac{A_0}{A_{k+1}}\right) f(x) + \frac{A_0}{A_{k+1}}\phi_0(x) = \left(1 - \frac{1}{A_{k+1}}\right) f(x) + \frac{1}{A_{k+1}}\phi_0(x).$$

Since the Lyapunov function property allows us to write

$$e^{\beta_t}\left(f(X_t) + \frac{\mu}{2}\|x - Z_t\|^2\right) \leq (e^{\beta_t} - e^{\beta_0}) f(x) + e^{\beta_0}\left(f(X_0) + \frac{\mu}{2}\|x - Z_0\|^2\right),$$

we can extract $\{f(X_t) + \frac{\mu}{2}\|x - Z_t\|^2, e^{\beta_t}\}$ *as the continuous-time estimate sequence for accelerated gradient descent in the strongly convex setting.*

B.7.4 Adagrad with momentum

We analyze Adagrad with momentum (2.84) *using the Lyapunov function* (2.85)*. Denote We check,*

$$\begin{aligned} \frac{E_{k+1} - E_k}{\delta} &= \frac{1}{2}\|x^* - z_{k+1}\|_{H_k}^2 - \frac{1}{2}\|x^* - z_k\|_{H_k}^2 + \frac{A_{k+1} - A_k}{\delta}(f(x_{k+1}) - f(x^*)) \\ & A_k \frac{f(x_{k+1}) - f(x_k)}{\delta} + \varepsilon_k^1 \\ &= -\left\langle H_k \frac{z_{k+1} - z_k}{\delta}, x^* - z_{k+1} \right\rangle + \frac{A_{k+1} - A_k}{\delta}(f(x_{k+1}) - f(x^*)) \\ & A_k \frac{f(x_{k+1}) - f(x_k)}{\delta} + \varepsilon_k^1 \\ &\overset{(2.84b)}{=} -\alpha_k D_f^g(x^*, x_{k+1}) - \alpha_k \langle g(x_{k+1}), x_{k+1} - z_{k+1} \rangle + A_k(f(x_{k+1}) - f(x_k)) + \varepsilon_k^1 \\ &\overset{(2.84b)}{=} -\alpha_k D_f^g(x^*, x_{k+1}) - A_k/\delta D_f^g(x_k, x_{k+1}) + \varepsilon_k^2 \end{aligned}$$

where the errors scale as $\varepsilon_k^1 = \frac{1}{2}\|x^* - z_{k+1}\|_{H_{k+1}}^2 - \frac{1}{2}\|x^* - z_{k+1}\|_{H_k}^2 - \frac{1}{\delta}\|z_{k+1} - z_k\|_{H_k}^2$ *and* $\varepsilon_k^2 = \varepsilon_k^1 + \alpha_k\langle \nabla f(x_{k+1}), z_k - z_{k+1}\rangle$. *We use Young's inequality to obtain the upper bound* $\varepsilon_k^2 \leq \frac{\alpha_k^2}{2\sigma}\|g(x_{k+1})\|_{H_k^*}^2 + \frac{1}{2}\|x^* - z_{k+1}\|_{H_{k+1}}^2 - \frac{1}{2}\|x^* - z_{k+1}\|_{H_k}^2$. *Using Theorem 7 and Lemma 4 in Duchi et al [16], we conclude the bounds* $\frac{1}{2}\|x^* - z_{k+1}\|_{H_{k+1}}^2 - \frac{1}{2}\|x^* - z_{k+1}\|_{H_k}^2 \leq \max_{k\leq K}\|x^* - z_k\|_2^2 tr(H_K^{1/2}) = D^2 tr(H_K^{1/2})$ *where* D *is the diameter of the set and* $\sum_{s=1}^K \|g(x_s)\|_{H_s^*}^2 \leq 2tr(H_K^{1/2})$. *From these upper bounds, we can conclude the an optimal* $f(x_k) - f(x^*) \leq O(1/\sqrt{k})$ *convergence rate. In particular, this method has a matching lower bound [16].*

Appendix C

Chapter Three

C.1 Preliminaries

Notation. *The notation is standard.* $[n] = \{1, 2, ..., n\}$ *refers to the set of integers from* 1 *to* n*, and* $2^{[n]}$ *refers to the set of all subsets of* $[n]$*. We let* $1_n \in \mathbb{R}^n$ *denote the vector of all ones. Given a square matrix* M *with real eigenvalues, we let* $\lambda_{\max}(M)$ *(resp.* $\lambda_{\min}(M)$*) denote the maximum (resp. minimum) eigenvalue of* M*. For two symmetric matrices* M, N*, the notation* $M \succcurlyeq N$ *(resp.* $M \succ N$*) means that the matrix* $M - N$ *is positive semi-definite (resp. positive definite). Every such* $M \succ 0$ *defines a real inner product space via the inner product* $\langle x, y \rangle_M = x^\mathsf{T} M y$*. We refer to its induced norm as* $\|x\|_M = \sqrt{\langle x, x \rangle_M}$*. The standard Euclidean inner product and norm will be denoted as* $\langle \cdot, \cdot \rangle$ *and* $\|\cdot\|_2$*, respectively. For an arbitrary matrix* M*, we let* $M^\dagger$ *denote its Moore-Penrose pseudo-inverse and* P_M *the orthogonal projector onto the range of* M*, which we denote as* $\mathcal{R}(M)$*. When* $M \succcurlyeq 0$*, we let* $M^{1/2}$ *denote its unique Hermitian square root. Finally, for a square* $n \times n$ *matrix* M*,* $\mathrm{diag}(M)$ *is the* $n \times n$ *diagonal matrix which contains the diagonal elements of* M*.*

Partitions on $[n]$**.** *In what follows, unless stated otherwise, whenever we discuss a partition of* $[n]$ *we assume that the partition is given by* $\bigcup_{i=1}^{n/p} J_i$*, where*

$$J_1 = \{1, 2, ..., p\}\ ,\ \ J_2 = \{p+1, p+2, ..., 2p\},\ \ \ldots\ .$$

This is without loss of generality because for any arbitrary equal sized partition of $[n]$*, there exists a permutation matrix* Π *such that all our results apply by the change of variables* $A \leftarrow \Pi^\mathsf{T} A \Pi$ *and* $b \leftarrow \Pi^\mathsf{T} b$*.*

C.2 Proofs for Separation Results (Section 3.4.3.1)

C.2.1 Expectation calculations (Propositions 3.4.1 and 3.4.2)

Recall the family of $n \times n$ positive definite matrices $\mathscr{A}$ defined in (3.35) as

$$A_{\alpha,\beta} = \alpha I + \frac{\beta}{n} 1_n 1_n^\mathsf{T}, \quad \alpha > 0, \alpha + \beta > 0 . \tag{C.1}$$

We first gather some elementary formulas. By the matrix inversion lemma,

$$A_{\alpha,\beta}^{-1} = \left(\alpha I + \frac{\beta}{n} 1_n 1_n^\mathsf{T}\right)^{-1} = \alpha^{-1} I - \frac{\beta/n}{\alpha(\alpha+\beta)} 1_n 1_n^\mathsf{T} . \tag{C.2}$$

Furthermore, let $S \in \mathbb{R}^{n \times p}$ be any column selector matrix with no duplicate columns. We have again by the matrix inversion lemma

$$(S^\mathsf{T} A_{\alpha,\beta} S)^{-1} = \left(\alpha I + \frac{\beta}{n} 1_p 1_p^\mathsf{T}\right)^{-1} = \alpha^{-1} I - \frac{\beta/n}{\alpha(\alpha + \beta p/n)} 1_p 1_p^\mathsf{T} . \tag{C.3}$$

The fact that the right hand side is independent of S is the key property which makes our calculations possible. Indeed, we have that

$$S(S^\mathsf{T} A_{\alpha,\beta} S)^{-1} S^\mathsf{T} = \alpha^{-1} S S^\mathsf{T} - \frac{\beta/n}{\alpha(\alpha + \beta p/n)} S 1_p 1_p^\mathsf{T} S^\mathsf{T} . \tag{C.4}$$

With these formulas in hand, our next proposition gathers calculations for the case when S represents uniformly choosing p columns without replacement.

Proposition C.2.1. *Consider the family of $n \times n$ positive definite matrices $\{A_{\alpha,\beta}\}$ from (C.1). Fix any integer p such that $1 < p < n$. Let $S \in \mathbb{R}^{n \times p}$ denote a random column selector matrix where each column of S is chosen uniformly at random without replacement from $\{e_1, ..., e_n\}$. For any $A_{\alpha,\beta}$,*

$$\mathbb{E}[S(S^\mathsf{T} A_{\alpha,\beta} S)^{-1} S^\mathsf{T} A_{\alpha,\beta}] = p\frac{(n-1)\alpha + (p-1)\beta}{(n-1)(n\alpha + p\beta)} I + \frac{(n-p)p\beta}{n(n-1)(n\alpha+p\beta)} 1_n 1_n^\mathsf{T} , \tag{C.5}$$

$$\begin{aligned}\mathbb{E}[S(S^\mathsf{T} A_{\alpha,\beta} S)^{-1} S^\mathsf{T} G_{\alpha,\beta}^{-1} S(S^\mathsf{T} A_{\alpha,\beta} S)^{-1} S^\mathsf{T}] = &\left(\frac{1}{\alpha} - \frac{(n-p)^2\beta}{(n-1)((n-1)\alpha + (p-1)\beta)(n\alpha+p\beta)}\right) I \\ &+ \frac{(p-1)\beta(n\alpha(1-2n) + np(\alpha-\beta) + p\beta)}{(n-1)n\alpha((n-1)\alpha + (p-1)\beta)(n\alpha+p\beta)} 1_n 1_n^\mathsf{T} .\end{aligned} \tag{C.6}$$

Above, $G_{\alpha,\beta} = \mathbb{E}[S(S^\mathsf{T} A_{\alpha,\beta} S)^{-1} S^\mathsf{T}]$.

Proof: *First, we have the following elementary expectation calculations,*

$$\mathbb{E}[SS^\mathsf{T}] = \frac{p}{n} I \,, \tag{C.7}$$

$$\mathbb{E}[S1_p 1_p^\mathsf{T} S^\mathsf{T}] = \frac{p}{n}\left(1 - \frac{p-1}{n-1}\right) I + \frac{p}{n}\left(\frac{p-1}{n-1}\right) 1_n 1_n^\mathsf{T} \,, \tag{C.8}$$

$$\mathbb{E}[SS^\mathsf{T} 1_n 1_p^\mathsf{T} S^\mathsf{T}] = \mathbb{E}[S1_p 1_n^\mathsf{T} SS^\mathsf{T}] = \mathbb{E}[SS^\mathsf{T} 1_n 1_n^\mathsf{T} SS^\mathsf{T}] = \mathbb{E}[S1_p 1_p^\mathsf{T} S^\mathsf{T}] \,, \tag{C.9}$$

$$\mathbb{E}[S1_p 1_p^\mathsf{T} S^\mathsf{T} 1_n 1_n^\mathsf{T} S1_p 1_p^\mathsf{T} S^\mathsf{T}] = \frac{p^3}{n}\left(1 - \frac{p-1}{n-1}\right) I + \frac{p^3}{n}\left(\frac{p-1}{n-1}\right) 1_n 1_n^\mathsf{T} \,. \tag{C.10}$$

To compute $G_{\alpha,\beta}$, we simply plug (C.7) and (C.8) into (C.4). After simplification,

$$G_{\alpha,\beta} = \mathbb{E}[S(S^\mathsf{T} A_{\alpha,\beta} S)^{-1} S^\mathsf{T}] = \frac{p}{\alpha n}\left(1 - \frac{\beta/n}{\alpha + \beta p/n}\left(1 - \frac{p-1}{n-1}\right)\right) I - \frac{p}{n}\frac{p-1}{n-1}\frac{\beta/n}{\alpha(\alpha+\beta p/n)} 1_n 1_n^\mathsf{T} \,.$$

From this formula for $G_{\alpha,\beta}$, (C.5) follows immediately.

Our next goal is to compute $\mathbb{E}[S(S^\mathsf{T} A_{\alpha,\beta} S)^{-1} S^\mathsf{T} G_{\alpha,\beta}^{-1} S(S^\mathsf{T} A_{\alpha,\beta} S)^{-1} S^\mathsf{T}]$. To do this, we first invert $G_{\alpha,\beta}$. Applying the matrix inversion lemma, we can write down a formula for the inverse of $G_{\alpha,\beta}$,

$$G_{\alpha,\beta}^{-1} = \underbrace{\frac{(n-1)\alpha(n\alpha + p\beta)}{(n-1)p\alpha + (p-1)p\beta}}_{\gamma} I + \underbrace{\frac{(p-1)\beta(n\alpha+p\beta)}{np((n-1)\alpha + (p-1)\beta)}}_{\eta} 1_n 1_n^\mathsf{T} \,. \tag{C.11}$$

Next, we note for any r, q, using the properties that $S^\mathsf{T} S = I$, $1_n^\mathsf{T} S 1_p = p$, and $1_p^\mathsf{T} 1_p = p$, we have that

$$\begin{aligned}
&(rSS^\mathsf{T} + qS1_p 1_p^\mathsf{T} S^\mathsf{T})(\gamma I + \eta 1_n 1_n^\mathsf{T})(rSS^\mathsf{T} + qS1_p 1_p^\mathsf{T} S^\mathsf{T}) \\
&\quad = \gamma r^2 SS^\mathsf{T} + 2r\gamma q S1_p 1_p^\mathsf{T} S^\mathsf{T} + \eta r^2 SS^\mathsf{T} 1_n 1_n^\mathsf{T} SS^\mathsf{T} \\
&\qquad + pr\eta q(SS^\mathsf{T} 1_n 1_p^\mathsf{T} S^\mathsf{T} + S1_p 1_n^\mathsf{T} SS^\mathsf{T}) + pq^2 \gamma S1_p 1_p^\mathsf{T} S^\mathsf{T} \\
&\qquad + \eta q^2 S1_p 1_p^\mathsf{T} S^\mathsf{T} 1_n 1_n^\mathsf{T} S1_p 1_p^\mathsf{T} S^\mathsf{T} \,.
\end{aligned}$$

Taking expectations of both sides of the above equation and using the formulas in (C.7), (C.8), (C.9), and (C.10),

$$\begin{aligned}
&\mathbb{E}[(rSS^\mathsf{T} + qS1_p 1_p^\mathsf{T} S^\mathsf{T})(\gamma I + \eta 1_n 1_n^\mathsf{T})(rSS^\mathsf{T} + qS1_p 1_p^\mathsf{T} S^\mathsf{T})] \\
&\quad = \frac{p(p(n-p)q^2 + 2(n-p)qr + (n-1)r^2)\gamma + p(n-p)(pq+r)^2\eta}{n(n-1)} I \\
&\qquad + \frac{p(p-1)(q(pq+2r)\gamma + (pq+r)^2\eta)}{n(n-1)} 1_n 1_n^\mathsf{T} \,.
\end{aligned}$$

We now set $r = \alpha^{-1}$, $q = -\frac{\beta/n}{\alpha(\alpha+\beta p/n)}$, and γ, η from (C.11) to reach the desired formula for (C.6). Proposition 3.4.2 follows immediately from Proposition C.2.1 by plugging in $\alpha = 1$

into (C.5). We next consider how (C.4) behaves under a fixed partition of $\{1,...,n\}$. Recall our assumption on partitions: $n = pk$ for some integer $k \geq 1$, and we sequentially partition $\{1,...,n\}$ into k partitions of size p, i.e. $J_1 = \{1,...,p\}$, $J_2 = \{p+1,...,2p\}$, and so on. Define $S_1,...,S_k \in \mathbb{R}^{n\times p}$ such that S_i is the column selector matrix for the partition J_i, and S uniformly chooses S_i with probability $1/k$.

Proposition C.2.2. *Consider the family of $n \times n$ positive definite matrices $\{A_{\alpha,\beta}\}$ from (C.1), and let n, p, and S be described as in the preceding paragraph. We have that*

$$\mathbb{E}[S(S^\mathsf{T}A_{\alpha,\beta}S)^{-1}S^\mathsf{T}A_{\alpha,\beta}] = \frac{p}{n}I + \frac{p\beta}{n^2\alpha+np\beta}1_n1_n^\mathsf{T} - \frac{p\beta}{n^2\alpha+np\beta}\mathrm{blkdiag}(\underbrace{1_p1_p^\mathsf{T},...,1_p1_p^\mathsf{T}}_{k\ times}) . \tag{C.12}$$

Proof: *Once again, the expectation calculations are*

$$\mathbb{E}[SS^\mathsf{T}] = \frac{p}{n}I, \quad \mathbb{E}[S1_p1_p^\mathsf{T}S^\mathsf{T}] = \frac{p}{n}\mathrm{blkdiag}(\underbrace{1_p1_p^\mathsf{T},...,1_p1_p^\mathsf{T}}_{k\ times}) .$$

Therefore,

$$\mathbb{E}[S(S^\mathsf{T}A_{\alpha,\beta}S)^{-1}S^\mathsf{T}] = \frac{p}{\alpha n}I - \frac{p}{n}\frac{\beta/n}{\alpha(\alpha+\beta p/n)}\mathrm{blkdiag}(1_p1_p^\mathsf{T},...,1_p1_p^\mathsf{T}) .$$

Furthermore,

$$\mathrm{blkdiag}(1_p1_p^\mathsf{T},...,1_p1_p^\mathsf{T})1_n1_n^\mathsf{T} = 1_n1_n^\mathsf{T}\mathrm{blkdiag}(1_p1_p^\mathsf{T},...,1_p1_p^\mathsf{T}) = p1_n1_n^\mathsf{T} ,$$

Hence, the formula for $\mathbb{E}[S(S^\mathsf{T}A_{\alpha,\beta}S)^{-1}S^\mathsf{T}A_{\alpha,\beta}]$ follows.

We now make the following observation. Let $Q_1,...,Q_k$ be any partition of $\{1,...,n\}$ into k partitions of size p. Let $\mathbb{E}_{S\sim Q_i}$ denote expectation with respect to S uniformly chosen as column selectors among $Q_1,...,Q_k$, and let $\mathbb{E}_{S\sim J_i}$ denote expectation with respect to the S in the setting of Proposition C.2.2. It is not hard to see there exists a permutation matrix Π such that

$$\Pi^\mathsf{T}\mathbb{E}_{S\sim Q_i}[S(S^\mathsf{T}A_{\alpha,\beta}S)^{-1}S^\mathsf{T}]\Pi = \mathbb{E}_{S\sim J_i}[S(S^\mathsf{T}A_{\alpha,\beta}S)^{-1}S^\mathsf{T}] .$$

Using this permutation matrix Π,

$$\begin{aligned}
\lambda_{\min}(\mathbb{E}_{S\sim Q_i}[P_{A_{\alpha,\beta}^{1/2}S}]) &= \lambda_{\min}(\mathbb{E}_{S\sim Q_i}[S(S^\mathsf{T}A_{\alpha,\beta}S)^{-1}S^\mathsf{T}]A_{\alpha,\beta}) \\
&= \lambda_{\min}(\mathbb{E}_{S\sim Q_i}[S(S^\mathsf{T}A_{\alpha,\beta}S)^{-1}S^\mathsf{T}]\Pi A_{\alpha,\beta}\Pi^\mathsf{T}) \\
&= \lambda_{\min}(\Pi^\mathsf{T}\mathbb{E}_{S\sim Q_i}[S(S^\mathsf{T}A_{\alpha,\beta}S)^{-1}S^\mathsf{T}]\Pi A_{\alpha,\beta}) \\
&= \lambda_{\min}(\mathbb{E}_{S\sim J_i}[S(S^\mathsf{T}A_{\alpha,\beta}S)^{-1}S^\mathsf{T}]A_{\alpha,\beta}) \\
&= \lambda_{\min}(\mathbb{E}_{S\sim J_i}[P_{A_{\alpha,\beta}^{1/2}S}]) .
\end{aligned}$$

Above, the second equality holds because $A_{\alpha,\beta}$ is invariant under a similarity transform by any permutation matrix. Therefore, Proposition C.2.2 yields the μ_{part} value for every partition $Q_1,...,Q_k$. The claim of Proposition 3.4.1 now follows by substituting $\alpha = 1$ into (C.12).

C.2.2 Proof of Proposition 3.4.3

Define $e_k = x_k - x_*$, $H_k = S_k(S_k^\mathsf{T} A S_k)^\dagger S_k^\mathsf{T}$ *and* $G = \mathbb{E}[H_k]$. *From the update rule* (3.19),

$$e_{k+1} = (I - H_k A)e_k \Longrightarrow A^{1/2}e_{k+1} = (I - A^{1/2}H_k A^{1/2})A^{1/2}e_k \,.$$

Taking and iterating expectations,

$$\mathbb{E}[A^{1/2}e_{k+1}] = (I - A^{1/2}GA^{1/2})\mathbb{E}[A^{1/2}e_k] \,.$$

Unrolling this recursion yields for all $k \geq 0$,

$$\mathbb{E}[A^{1/2}e_k] = (I - A^{1/2}GA^{1/2})^k A^{1/2}e_0 \,.$$

Choose $A^{1/2}e_0 = v$, *where* v *is an eigenvector of* $I - A^{1/2}GA^{1/2}$ *with eigenvalue* $\lambda_{\max}(I - A^{1/2}GA^{1/2}) = 1 - \lambda_{\min}(GA) = 1 - \mu$. *Now by Jensen's inequality,*

$$\mathbb{E}[\|e_k\|_A] = \mathbb{E}[\|A^{1/2}e_k\|_2] \geq \|\mathbb{E}[A^{1/2}e_k]\|_2 = (1-\mu)^k \|e_0\|_A \,.$$

This establishes the claim.

C.3 Proofs for Convergence Results (Section 3.4.3.2)

We now state our main structural result for accelerated coordinate descent. Let $\mathbb{P}$ *be a probability measure on* $\Omega = \mathcal{S}^{n\times n} \times \mathbb{R}_+ \times \mathbb{R}_+$, *with* $\mathcal{S}^{n\times n}$ *denoting* $n \times n$ *positive semi-definite matrices and* $\mathbb{R}_+$ *denoting positive reals. Write* $\omega \in \Omega$ *as the tuple* $\omega = (H, \Gamma, \gamma)$, *and let* $\mathbb{E}$ *denote expectation with respect to* $\mathbb{P}$. *Suppose that* $G = \mathbb{E}[\frac{1}{\gamma}H]$ *exists and is positive definite.*

Now suppose that $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ *is a differentiable and strongly convex function, and put* $f_* = \min_x f(x)$, *with* x_* *attaining the minimum value. Suppose that* f *is both* μ*-strongly convex and has* L*-Lipschitz gradients with respect to the* G^{-1} *norm. This means that for all* $x, y \in \mathbb{R}^n$, *we have*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle + \frac{\mu}{2}\|y - x\|^2_{G^{-1}} \,, \tag{C.13a}$$

$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}\|y - x\|^2_{G^{-1}} \,. \tag{C.13b}$$

We now define a random sequence as follows. Let $\omega_0 = (H_0, \Gamma_0, \gamma_0), \omega_1 = (H_1, \Gamma_1, \gamma_1), \ldots$ *be independent realizations from* $\mathbb{P}$. *Starting from* $y_0 = z_0 = x_0$ *with* x_0 *fixed, consider the sequence* $\{(x_k, y_k, z_k)\}_{k\geq 0}$ *defined by the recurrence*

$$\tau(x_{k+1} - z_k) = y_k - x_{k+1} \,, \tag{C.14a}$$

$$y_{k+1} = x_{k+1} - \frac{1}{\Gamma_k}H_k \nabla f(x_{k+1}) \,, \tag{C.14b}$$

$$z_{k+1} - z_k = \tau\left(x_{k+1} - z_k - \frac{1}{\mu\gamma_k}H_k \nabla f(x_{k+1})\right) \,. \tag{C.14c}$$

It is easily verified that $(x, y, z) = (x_*, x_*, x_*)$ is a fixed point of the aforementioned dynamical system. Our goal for now is to describe conditions on f, μ, and τ such that the sequence of updates (C.14a), (C.14b), and (C.14c) converges to this fixed point. As described in Wilson et al. [77], our main strategy for proving convergence will be to introduce the following Lyapunov function

$$E_k = f(y_k) - f_* + \frac{\mu}{2}\|z_k - x_*\|^2_{G^{-1}} \,, \tag{C.15}$$

and show that E_k decreases along every trajectory. We let $\mathbb{E}_k$ denote the expectation conditioned on $\mathcal{F}_k = \sigma(\omega_0, \omega_1, ..., \omega_{k-1})$. Observe that x_{k+1} is $\mathcal{F}_k$-measurable, a fact we will use repeatedly throughout our calculations. With the preceding definitions in place, we state and prove our main structural theorem.

Theorem C.3.1. *(Generalization of Theorem 3.4.4.) Let f and G be as defined above, with f satisfying μ-strongly convexity and L-Lipschitz gradients with respect to the $\|\cdot\|_{G^{-1}}$ norm, as defined in (C.13a) and (C.13b). Suppose that for all fixed $x \in \mathbb{R}^n$, we have that the following holds for almost every $\omega \in \Omega$,*

$$f(\Phi(x;\omega)) \leq f(x) - \frac{1}{2\Gamma}\|\nabla f(x)\|^2_H \,, \;\; \Phi(x;\omega) = x - \frac{1}{\Gamma}H\nabla f(x) \,. \tag{C.16}$$

Furthermore, suppose that $\nu > 0$ satisfies

$$\mathbb{E}\left[\frac{1}{\gamma^2}HG^{-1}H\right] \preccurlyeq \nu\mathbb{E}\left[\frac{1}{\gamma^2}H\right] \,. \tag{C.17}$$

Then as long as we set $\tau > 0$ such that τ satisfies for almost every $\omega \in \Omega$,

$$\tau \leq \frac{\gamma}{\sqrt{\Gamma}}\sqrt{\frac{\mu}{\nu}} \,, \;\; \tau \leq \sqrt{\frac{\mu}{L}} \,, \tag{C.18}$$

we have that E_k defined in (C.15) satisfies for all $k \geq 0$,

$$\mathbb{E}_k[E_{k+1}] \leq (1-\tau)E_k \,. \tag{C.19}$$

Proof. First, recall the following two point equality valid for any vectors $a, b, c \in V$ in a real inner product space V,

$$\|a-b\|^2_V - \|c-b\|^2_V = \|a-c\|^2_V + 2\langle a-c, c-b\rangle_V \,. \tag{C.20}$$

Now we can proceed with our analysis,

$$
\begin{aligned}
E_{k+1} - E_k &\overset{(C.20)}{=} f(y_{k+1}) - f(y_k) - \mu\langle z_{k+1} - z_k, x_* - z_k\rangle_{G^{-1}} + \frac{\mu}{2}\|z_{k+1} - z_k\|^2_{G^{-1}} \\
&= f(y_{k+1}) - f(x_{k+1}) + f(x_{k+1}) - f(y_k) - \mu\langle z_{k+1} - z_k, x_* - z_k\rangle_{G^{-1}} + \frac{\mu}{2}\|z_{k+1} - z_k\|^2_{G^{-1}} \\
&\overset{(C.13a)}{\leq} f(y_{k+1}) - f(x_{k+1}) + \langle \nabla f(x_{k+1}), x_{k+1} - y_k\rangle - \frac{\mu}{2}\|x_{k+1} - y_k\|^2_{G^{-1}} \\
&\quad - \mu\langle z_{k+1} - z_k, x_* - z_k\rangle_{G^{-1}} + \frac{\mu}{2}\|z_{k+1} - z_k\|^2_{G^{-1}} \qquad \text{(C.21a)} \\
&\overset{(C.14c)}{=} f(y_{k+1}) - f(x_{k+1}) + \langle \nabla f(x_{k+1}), x_{k+1} - y_k\rangle - \frac{\mu}{2}\|x_{k+1} - y_k\|^2_{G^{-1}} \\
&\quad + \tau\langle \frac{1}{\gamma_k} H_k \nabla f(x_{k+1}) - \mu(x_{k+1} - z_k), x_* - z_k\rangle_{G^{-1}} + \frac{\mu}{2}\|z_{k+1} - z_k\|^2_{G^{-1}} \qquad \text{(C.21b)} \\
&= f(y_{k+1}) - f(x_{k+1}) + \langle \nabla f(x_{k+1}), x_{k+1} - y_k\rangle - \frac{\mu}{2}\|x_{k+1} - y_k\|^2_{G^{-1}} \\
&\quad + \tau\langle \frac{1}{\gamma_k} H_k \nabla f(x_{k+1}), x_* - x_{k+1}\rangle_{G^{-1}} + \tau\langle \frac{1}{\gamma_k} H_k \nabla f(x_{k+1}), x_{k+1} - z_k\rangle_{G^{-1}} \\
&\quad - \tau\mu\langle x_{k+1} - z_k, x_* - z_k\rangle_{G^{-1}} + \frac{\mu}{2}\|z_{k+1} - z_k\|^2_{G^{-1}} \\
&\overset{(C.14c)}{=} f(y_{k+1}) - f(x_{k+1}) + \langle \nabla f(x_{k+1}), x_{k+1} - y_k\rangle - \frac{\mu}{2}\|x_{k+1} - y_k\|^2_{G^{-1}} \\
&\quad + \tau\langle \frac{1}{\gamma_k} H_k \nabla f(x_{k+1}), x_* - x_{k+1}\rangle_{G^{-1}} + \tau\langle \frac{1}{\gamma_k} H_k \nabla f(x_{k+1}), x_{k+1} - z_k\rangle_{G^{-1}} \\
&\quad - \tau\mu\langle x_{k+1} - z_k, x_* - z_k\rangle_{G^{-1}} + \frac{\mu}{2}\|\tau(x_{k+1} - z_k)\|^2_{G^{-1}} + \frac{\tau^2}{2\mu\gamma_k^2}\|H_k \nabla f(x_{k+1})\|^2_{G^{-1}} \\
&\quad - \tau\langle x_{k+1} - z_k, \tau\frac{1}{\gamma_k} H_k \nabla f(x_{k+1})\rangle_{G^{-1}} \qquad \text{(C.21c)} \\
&\overset{(C.16)}{\leq} -\frac{1}{2\Gamma_k}\|\nabla f(x_{k+1})\|^2_{H_k} + \langle \nabla f(x_{k+1}), x_{k+1} - y_k\rangle - \frac{\mu}{2}\|x_{k+1} - y_k\|^2_{G^{-1}} \\
&\quad + \tau\langle \frac{1}{\gamma_k} H_k \nabla f(x_{k+1}), x_* - x_{k+1}\rangle_{G^{-1}} + \tau\langle \frac{1}{\gamma_k} H_k \nabla f(x_{k+1}), x_{k+1} - z_k\rangle_{G^{-1}} \\
&\quad - \tau\mu\langle x_{k+1} - z_k, x_* - z_k\rangle_{G^{-1}} + \frac{\mu}{2}\|\tau(x_{k+1} - z_k)\|^2_{G^{-1}} + \frac{\tau^2}{2\mu\gamma_k^2}\|H_k \nabla f(x_{k+1})\|^2_{G^{-1}} \\
&\quad - \tau\langle x_{k+1} - z_k, \tau\frac{1}{\gamma_k} H_k \nabla f(x_{k+1})\rangle_{G^{-1}} \,. \qquad \text{(C.21d)}
\end{aligned}
$$

Above, (C.21a) follows from μ-strong convexity, (C.21b) and (C.21c) both use the definition of the update sequence given in (C.14), and (C.21d) follows using the gradient inequality

assumption (C.16). Now letting $x \in \mathbb{R}^n$ be fixed, we observe that

$$\mathbb{E}\left[\frac{\tau^2}{2\mu\gamma^2}\nabla f(x)^\intercal H G^{-1} H \nabla f(x) - \frac{1}{2\Gamma}\|\nabla f(x)\|_H^2\right] \overset{(C.17)}{\leq} \mathbb{E}\left[\left(\frac{\tau^2\nu}{2\mu\gamma^2} - \frac{1}{2\Gamma}\right)\|\nabla f(x)\|_H^2\right] \overset{(C.18)}{\leq} 0\,. \tag{C.22}$$

The first inequality uses the assumption on ν, and the second inequality uses the requirement

that $\tau \leq \frac{\gamma}{\sqrt{\Gamma}}\sqrt{\frac{\mu}{\nu}}$. Now taking expectations with respect to $\mathbb{E}_k$,

$$\begin{aligned}
\mathbb{E}_k[E_{k+1}] - E_k \leq\ & \mathbb{E}_k\left[\frac{\tau^2}{2\mu\gamma_k^2}\nabla f(x_{k+1})^\mathsf{T} H_k G^{-1} H_k \nabla f(x_{k+1}) - \frac{1}{2\Gamma_k}\|\nabla f(x_{k+1})\|_{H_k}^2\right] \\
& + \langle \nabla f(x_{k+1}), x_{k+1} - y_k\rangle - \frac{\mu}{2}\|x_{k+1} - y_k\|_{G^{-1}}^2 \\
& + \tau\langle \nabla f(x_{k+1}), x_* - x_{k+1}\rangle + \tau\langle \nabla f(x_{k+1}), x_{k+1} - z_k\rangle - \tau\mu\langle x_{k+1} - z_k, x_* - z_k\rangle_{G^{-1}} \\
& + \frac{\mu}{2}\|\tau(x_{k+1} - z_k)\|_{G^{-1}}^2 - \tau\langle x_{k+1} - z_k, \tau\nabla f(x_{k+1})\rangle \\
\overset{(C.22)}{\leq}\ & \langle \nabla f(x_{k+1}), x_{k+1} - y_k\rangle - \frac{\mu}{2}\|x_{k+1} - y_k\|_{G^{-1}}^2 + \tau\langle \nabla f(x_{k+1}), x_* - x_{k+1}\rangle \\
& + \tau\langle \nabla f(x_{k+1}), x_{k+1} - z_k\rangle - \tau\mu\langle x_{k+1} - z_k, x_* - z_k\rangle_{G^{-1}} \\
& + \frac{\mu}{2}\|\tau(x_{k+1} - z_k)\|_{G^{-1}}^2 - \tau\langle x_{k+1} - z_k, \tau\nabla f(x_{k+1})\rangle \\
\overset{(C.13a)}{\leq}\ & -\tau\left(f(x_{k+1}) - f_* + \frac{\mu}{2}\|x_{k+1} - x_*\|_{G^{-1}}^2\right) + \langle \nabla f(x_{k+1}), x_{k+1} - y_k\rangle - \frac{\mu}{2}\|x_{k+1} - y_k\|_{G^{-1}}^2 \\
& + \tau\langle \nabla f(x_{k+1}), x_{k+1} - z_k\rangle - \tau\mu\langle x_{k+1} - z_k, x_* - z_k\rangle_{G^{-1}} \\
& + \frac{\mu}{2}\|\tau(x_{k+1} - z_k)\|_{G^{-1}}^2 - \tau\langle x_{k+1} - z_k, \tau\nabla f(x_{k+1})\rangle \qquad \text{(C.23a)} \\
\overset{(C.14a)}{=}\ & -\tau\left(f(x_{k+1}) - f_* + \frac{\mu}{2}\|x_{k+1} - x_*\|_{G^{-1}}^2\right) - \frac{\mu}{2}\|x_{k+1} - y_k\|_{G^{-1}}^2 \\
& - \tau\mu\langle x_{k+1} - z_k, x_* - z_k\rangle_{G^{-1}} \\
& + \frac{\mu}{2}\|\tau(x_{k+1} - z_k)\|_{G^{-1}}^2 - \tau\langle y_k - x_{k+1}, \nabla f(x_{k+1})\rangle \qquad \text{(C.23b)} \\
\overset{(C.13b)}{\leq}\ & -\tau\left(f(x_{k+1}) - f_* + \frac{\mu}{2}\|x_{k+1} - x_*\|_{G^{-1}}^2\right) - \frac{\mu}{2}\|x_{k+1} - y_k\|_{G^{-1}}^2 \\
& - \tau\mu\langle x_{k+1} - z_k, x_* - z_k\rangle_{G^{-1}} \\
& + \frac{\mu}{2}\|\tau(x_{k+1} - z_k)\|_{G^{-1}}^2 + \tau(f(x_{k+1}) - f(y_k)) + \frac{\tau L}{2}\|y_k - x_{k+1}\|_{G^{-1}}^2 \\
& \qquad \text{(C.23c)} \\
\overset{(C.20)}{=}\ & -\tau\left(f(x_{k+1}) - f_* + \frac{\mu}{2}\|x_{k+1} - z_k\|_{G^{-1}}^2 + \frac{\mu}{2}\|z_k - x_*\|_{G^{-1}}^2 + \mu\langle x_{k+1} - z_k, z_k - x_*\rangle_{G^{-1}}\right) \\
& - \frac{\mu}{2}\|x_{k+1} - y_k\|_{G^{-1}}^2 - \tau\mu\langle x_{k+1} - z_k, x_* - z_k\rangle_{G^{-1}} \\
& + \frac{\mu}{2}\|\tau(x_{k+1} - z_k)\|_{G^{-1}}^2 + \tau(f(x_{k+1}) - f(y_k)) + \frac{\tau L}{2}\|y_k - x_{k+1}\|_{G^{-1}}^2 \\
& \qquad \text{(C.23d)} \\
\overset{(C.15)}{=}\ & -\tau E_k - \frac{\mu}{2}\|x_{k+1} - y_k\|_{G^{-1}}^2 - \frac{\tau\mu}{2}\|x_{k+1} - z_k\|_{G^{-1}}^2 \\
& + \frac{\mu}{2}\|\tau(x_{k+1} - z_k)\|_{G^{-1}}^2 + \frac{\tau L}{2}\|y_k - x_{k+1}\|_{G^{-1}}^2 \\
\overset{(C.14a)}{=}\ & -\tau E_k + \left(\frac{\tau L}{2} - \frac{\mu}{2\tau}\right)\|y_k - x_{k+1}\|_{G^{-1}}^2 \qquad \text{(C.23e)} \\
\overset{(C.18)}{\leq}\ & -\tau E_k\,.
\end{aligned}$$

Above, (C.23a) follows from μ-strong convexity, (C.23b) and (C.23e) both use the definition of the sequence (C.14), (C.23c) follows from L-Lipschitz gradients, (C.23d) uses the two-point inequality (C.20), and the last inequality follows from the assumption of $\tau \leq \sqrt{\frac{\mu}{L}}$. The claim (C.19) now follows by re-arrangement. □

C.3.1 Proof of Theorem 3.4.5

Next, we describe how to recover Theorem 3.4.5 from Theorem C.3.1. We do this by applying Theorem C.3.1 to the function $f(x) = \frac{1}{2}x^\mathsf{T}Ax - x^\mathsf{T}b$.

The first step in applying Theorem C.3.1 is to construct a probability measure on $\mathcal{S}^{n\times n} \times \mathbb{R}_+ \times \mathbb{R}_+$ *for which the randomness of the updates is drawn from. We already have a distribution on* $\mathcal{S}^{n\times n}$ *from setting of Theorem 3.4.5 via the random matrix* H. *We trivially augment this distribution by considering the random variable* $(H, 1, 1) \in \Omega$. *By setting* $\Gamma = \gamma = 1$, *the sequence (C.14a), (C.14b), (C.14c) reduces to that of Algorithm 1. Furthermore, the requirement on the* ν *parameter from (C.17) simplifies to the requirement listed in (3.31). This holds by the following equivalences which are valid since conjugation by* G *(which is assumed to be positive definite) preserves the semi-definite ordering,*

$$\begin{aligned}\lambda_{\max}\left(\mathbb{E}\left[(G^{-1/2}HG^{-1/2})^2\right]\right) \leq \nu &\iff \mathbb{E}\left[(G^{-1/2}HG^{-1/2})^2\right] \preccurlyeq \nu I \\ &\iff \mathbb{E}\left[G^{-1/2}HG^{-1}HG^{-1/2}\right] \preccurlyeq \nu I \\ &\iff \mathbb{E}\left[HG^{-1}H\right] \preccurlyeq \nu G\,. \end{aligned} \tag{C.24}$$

It remains to check the gradient inequality (C.16) and compute the strong convexity and Lipschitz parameters. These computations fall directly from the calculations made in Theorem 1 of [58], but we replicate them here for completeness.

To check the gradient inequality (C.16), because f *is a quadratic function, its second order Taylor expansion is exact. Hence for almost every* $\omega \in \Omega$,

$$\begin{aligned} f(\Phi(x;\omega)) &= f(x) - \langle \nabla f(x), H\nabla f(x)\rangle + \frac{1}{2}\nabla f(x)^\mathsf{T} HAH\nabla f(x) \\ &= f(x) - \langle \nabla f(x), H\nabla f(x)\rangle + \frac{1}{2}\nabla f(x)^\mathsf{T} S(S^\mathsf{T}AS)^\dagger S^\mathsf{T}AS(S^\mathsf{T}AS)^\dagger S^\mathsf{T}\nabla f(x) \\ &= f(x) - \langle \nabla f(x), H\nabla f(x)\rangle + \frac{1}{2}\nabla f(x)^\mathsf{T} S(S^\mathsf{T}AS)^\dagger S^\mathsf{T}\nabla f(x) \\ &= f(x) - \frac{1}{2}\nabla f(x)^\mathsf{T} H\nabla f(x)\,. \end{aligned}$$

Hence the inequality (C.16) holds with equality.

We next compute the strong convexity and Lipschitz gradient parameters. We first show that f *is* $\lambda_{\min}(\mathbb{E}[P_{A^{1/2}S}])$*-strongly convex with respect to the* $\|\cdot\|_{G^{-1}}$ *norm. This follows since*

for any $x, y \in \mathbb{R}^n$, using the assumption that G is positive definite,

$$\begin{aligned} f(y) &= f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2}(y-x)^\mathsf{T} A (y-x) \\ &= f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2}(y-x)^\mathsf{T} G^{-1/2} G^{1/2} A G^{1/2} G^{-1/2} (y-x) \\ &\geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\lambda_{\min}(A^{1/2} G A^{1/2})}{2} \|y - x\|_{G^{-1}}^2 . \end{aligned}$$

The strong convexity bound now follows since

$$A^{1/2} G A^{1/2} = A^{1/2} \mathbb{E}[H] A^{1/2} = \mathbb{E}[A^{1/2} S (S^\mathsf{T} A S)^\dagger S^\mathsf{T} A^{1/2}] = \mathbb{E}[P_{A^{1/2} S}] .$$

An nearly identical argument shows that f is $\lambda_{\max}(\mathbb{E}[P_{A^{1/2}S}])$-strongly convex with respect to the $\|\cdot\|_{G^{-1}}$ norm. Since the eigenvalues of projector matrices are bounded by 1, we have that f is 1-Lipschitz with respect to the $\|\cdot\|_{G^{-1}}$ norm. This calculation shows that the requirement on τ from (C.18) simplifies to $\tau \leq \sqrt{\frac{\mu}{\nu}}$, since $L = 1$ and $\nu \geq 1$ by Proposition C.6.1 which we state and prove later.

At this point, Theorem C.3.1 yields that $\mathbb{E}[E_k] \leq (1-\tau)^k E_0$. To recover the final claim (3.32), recall that $f(y_k) - f_* = \frac{1}{2}\|y_k - x_*\|_A^2$. Furthermore, $\mu G^{-1} \preccurlyeq A$, since

$$\begin{aligned} \mu \leq \lambda_{\min}(A^{1/2} G A^{1/2}) &\iff \mu \leq \lambda_{\min}(G^{1/2} A G^{1/2}) \\ &\iff \mu I \preccurlyeq G^{1/2} A G^{1/2} \\ &\iff \mu G^{-1} \preccurlyeq A . \end{aligned}$$

Hence, we can upper bound E_0 as follows

$$\begin{aligned} E_0 &= f(y_0) - f_* + \frac{\mu}{2}\|z_0 - x_*\|_{G^{-1}}^2 = \frac{1}{2}\|y_0 - x_*\|_A^2 + \frac{\mu}{2}\|z_0 - x_*\|_{G^{-1}}^2 \\ &\leq \frac{1}{2}\|y_0 - x_*\|_A^2 + \frac{1}{2}\|z_0 - x_*\|_A^2 = \|x_0 - x_*\|_A^2 . \end{aligned}$$

On the other hand, we have that $\frac{1}{2}\|y_k - x_*\|_A^2 \leq E_k$. Putting the inequalities together,

$$\frac{1}{\sqrt{2}}\mathbb{E}[\|y_k - x_*\|_A] \leq \sqrt{\mathbb{E}[\frac{1}{2}\|y_k - x_*\|_A^2]} \leq \sqrt{\mathbb{E}[E_k]} \leq \sqrt{(1-\tau)^k E_0} \leq (1-\tau)^{k/2}\|x_0 - x_*\|_A^2 ,$$

where the first inequality holds by Jensen's inequality. The claimed inequality (3.32) now follows.

C.3.2 Proof of Proposition 3.4.6

We first state and prove an elementary linear algebra fact which we will use below in our calculations.

Proposition C.3.2. *Let A, B, C, D be $n \times n$ diagonal matrices, and define $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$. The eigenvalues of M are given by the union of the eigenvalues of the 2×2 matrices*

$$\begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}, \quad i = 1, ..., n,$$

where A_i, B_i, C_i, D_i denote the i-th diagonal entry of A, B, C, D respectively.

Proof. For every $s \in \mathbb{C}$ we have that the matrices $-C$ and $sI - D$ are diagonal and hence commute. Applying the corresponding formula for a block matrix determinant under this assumption,

$$\begin{aligned} 0 &= \det \begin{bmatrix} sI - A & -B \\ -C & sI - D \end{bmatrix} = \det((sI - A)(sI - D) - BC) \\ &= \prod_{i=1}^{n}((s - A_i)(s - D_i) - B_iC_i) = \prod_{i=1}^{n} \det \begin{bmatrix} s - A_i & -B_i \\ -C_i & s - D_i \end{bmatrix}. \end{aligned}$$

□

Now we proceed with the proof of Proposition 3.4.6. Define $e_k = \begin{bmatrix} y_k - x_* \\ z_k - x_* \end{bmatrix}$. It is easy to see from the definition of Algorithm 1 that $\{e_k\}$ satisfies the recurrence

$$e_{k+1} = \frac{1}{1+\tau} \begin{bmatrix} I - H_kA & \tau(I - H_kA) \\ \tau(I - \frac{1}{\mu}H_kA) & I - \frac{\tau^2}{\mu}H_kA \end{bmatrix} e_k .$$

Hence,

$$\begin{aligned} &\begin{bmatrix} A^{1/2} & 0 \\ 0 & \mu^{1/2}G^{-1/2} \end{bmatrix} e_{k+1} \\ &\quad = \frac{1}{1+\tau} \begin{bmatrix} A^{1/2} & 0 \\ 0 & \mu^{1/2}G^{-1/2} \end{bmatrix} \begin{bmatrix} I - H_kA & \tau(I - H_kA) \\ \tau(I - \frac{1}{\mu}H_kA) & I - \frac{\tau^2}{\mu}H_kA \end{bmatrix} e_k \\ &\quad = \frac{1}{1+\tau} \begin{bmatrix} A^{1/2} - A^{1/2}H_kA & \tau(A^{1/2} - A^{1/2}H_kA) \\ \mu^{1/2}\tau G^{-1/2}(I - \frac{1}{\mu}H_kA) & \mu^{1/2}G^{-1/2}(I - \frac{\tau^2}{\mu}H_kA) \end{bmatrix} e_k \\ &\quad = \frac{1}{1+\tau} \begin{bmatrix} I - A^{1/2}H_kA^{1/2} & \mu^{-1/2}\tau(A^{1/2} - A^{1/2}H_kA)G^{1/2} \\ \mu^{1/2}\tau G^{-1/2}(I - \frac{1}{\mu}H_kA)A^{-1/2} & G^{-1/2}(I - \frac{\tau^2}{\mu}H_kA)G^{1/2} \end{bmatrix} \begin{bmatrix} A^{1/2} & 0 \\ 0 & \mu^{1/2}G^{-1/2} \end{bmatrix} e_k . \end{aligned}$$

Define $P = \begin{bmatrix} A & 0 \\ 0 & \mu G^{-1} \end{bmatrix}$. By taking and iterating expectations,

$$\mathbb{E}[P^{1/2}e_{k+1}] = \frac{1}{1+\tau} \begin{bmatrix} I - A^{1/2}GA^{1/2} & \mu^{-1/2}\tau(A^{1/2}G^{1/2} - A^{1/2}GAG^{1/2}) \\ \mu^{1/2}\tau(G^{-1/2}A^{-1/2} - \frac{1}{\mu}G^{1/2}A^{1/2}) & I - \frac{\tau^2}{\mu}G^{1/2}AG^{1/2} \end{bmatrix} \mathbb{E}[P^{1/2}e_k] .$$

Denote the matrix $Q = A^{1/2}G^{1/2}$. *Unrolling the recurrence above yields that*

$$\mathbb{E}[P^{1/2}e_k] = R^k P^{1/2} e_0 \,, \;\; R = \frac{1}{1+\tau}\begin{bmatrix} I - QQ^\mathsf{T} & \mu^{-1/2}\tau(Q - QQ^\mathsf{T}Q) \\ \mu^{1/2}\tau(Q^{-1} - \frac{1}{\mu}Q^\mathsf{T}) & I - \frac{\tau^2}{\mu}Q^\mathsf{T}Q \end{bmatrix} .$$

Write the SVD of Q *as* $Q = U\Sigma V^\mathsf{T}$. *Both* U *and* V *are* $n \times n$ *orthonormal matrices. It is easy to see that* R^k *is given by*

$$R^k = \frac{1}{(1+\tau)^k}\begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix}\begin{bmatrix} I - \Sigma^2 & \mu^{-1/2}\tau(\Sigma - \Sigma^3) \\ \mu^{1/2}\tau(\Sigma^{-1} - \frac{1}{\mu}\Sigma) & I - \frac{\tau^2}{\mu}\Sigma^2 \end{bmatrix}^k \begin{bmatrix} U^\mathsf{T} & 0 \\ 0 & V^\mathsf{T} \end{bmatrix} . \tag{C.25}$$

Suppose we choose $P^{1/2}e_0$ *to be a right singular vector of* R^k *corresponding to the maximum singular value* $\sigma_{\max}(R^k)$. *Then we have that*

$$\mathbb{E}[\|P^{1/2}e_k\|_2] \geq \|\mathbb{E}[P^{1/2}e_k]\|_2 = \|R^k P^{1/2} e_0\|_2 = \sigma_{\max}(R^k)\|P^{1/2}e_0\|_2 \geq \rho(R^k)\|P^{1/2}e_0\|_2 \,,$$

where $\rho(\cdot)$ *denotes the spectral radius. The first inequality is Jensen's inequality, and the second inequality uses the fact that the spectral radius is bounded above by any matrix norm. The eigenvalues of* R^k *are the* k*-th power of the eigenvalues of* R *which, using the similarity transform* (C.25) *along with Proposition C.3.2, are given by the eigenvalues of the* 2×2 *matrices* R_i *defined as*

$$R_i = \frac{1}{1+\tau}\begin{bmatrix} 1 - \sigma_i^2 & \mu^{-1/2}\tau(\sigma_i - \sigma_i^3) \\ \mu^{1/2}\tau(\sigma_i^{-1} - \frac{1}{\mu}\sigma_i) & 1 - \frac{\tau^2}{\mu}\sigma_i^2 \end{bmatrix} , \;\; \sigma_i = \Sigma_{ii} \,, \;\; i = 1, ..., n \,.$$

On the other hand, since the entries in Σ *are given by the eigenvalues of* $A^{1/2}G^{1/2}G^{1/2}A^{1/2} = \mathbb{E}[P_{A^{1/2}S}]$, *there exists an* i *such that* $\sigma_i = \sqrt{\mu}$. *This* R_i *is upper triangular, and hence its eigenvalues can be read off the diagonal. This shows that* $\frac{1-\tau^2}{1+\tau} = 1 - \tau$ *is an eigenvalue of* R, *and hence* $(1-\tau)^k$ *is an eigenvalue of* R^k. *But this means that* $(1-\tau)^k \leq \rho(R^k)$. *Hence, we have shown that*

$$\mathbb{E}[\|P^{1/2}e_k\|_2] \geq (1-\tau)^k \|P^{1/2}e_0\|_2 \,.$$

The desired claim now follows from

$$\begin{aligned} \|P^{1/2}e_k\|_2 &= \sqrt{\|y_k - x_*\|_A^2 + \mu\|z_k - x_*\|_{G^{-1}}^2} \\ &\leq \sqrt{\|y_k - x_*\|_A^2 + \|z_k - x_*\|_A^2} \leq \|y_k - x_*\|_A + \|z_k - x_*\|_A \,, \end{aligned}$$

where the first inequality holds since $\mu G^{-1} \preccurlyeq A$ *and the second inequality holds since* $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ *for non-negative* a, b.

C.4 Recovering the ACDM Result from Nesterov and Stich [48]

We next show how to recover Theorem 1 of Nesterov and Stich [48] using Theorem C.3.1, in the case of $\alpha = 1$. *A nearly identical argument can also be used to recover the result of Allen-Zhu et al. [1] under the strongly convex setting in the case of* $\beta = 0$. *Our argument proceeds in two steps. First, we prove a convergence result for a simplified accelerated coordinate descent method which we introduce in Algorithm 2. Then, we describe how a minor tweak to ACDM shows the equivalence between ACDM and Algorithm 2.*

Before we proceed, we first describe the setting of Theorem 1. Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ *be a twice differentiable strongly convex function with Lipschitz gradients. Let* $J_1, ..., J_m$ *denote a partition of* $\{1, ..., n\}$ *into* m *partitions. Without loss of generality, we can assume that the partitions are in order, i.e.* $J_1 = \{1, ..., n_1\}$, $J_2 = \{n_1 + 1, ..., n_2\}$, *and so on. This is without loss of generality since we can always consider the function* $g(x) = f(\Pi x)$ *for a suitable permutation matrix* Π. *Let* $B_1, ..., B_m$ *be fixed positive definite matrices such that* $B_i \in \mathbb{R}^{|J_i| \times |J_i|}$. *Set* $H_i = S_i B_i^{-1} S_i^\mathsf{T}$, *where* $S_i \in \mathbb{R}^{n \times |J_i|}$ *is the column selector matrix associated to partition* J_i, *and define* $L_i = \sup_{x \in \mathbb{R}^n} \lambda_{\max}(B_i^{-1/2} S_i^\mathsf{T} \nabla^2 f(x) S_i B_i^{-1/2})$ *for* $i = 1, ..., m$. *Furthermore, define* $p_i = \frac{\sqrt{L_i}}{\sum_{j=1}^m \sqrt{L_j}}$.

C.4.1 Proof of convergence of a simplified accelerated coordinate descent method

Now consider the following accelerated randomized coordinate descent algorithm in Algorithm 2.

Theorem C.3.1 is readily applied to Algorithm 2 to give a convergence guarantee which matches the bound of Theorem 1 of Nesterov and Stich. We sketch the argument below.

Algorithm 2 instantiates (C.14) with the definitions above and particular choices $\Gamma_k = L_{i_k}$ *and* $\gamma_k = p_{i_k}$. *We will specify the choice of* μ *at a later point. To see that this setting is valid, we construct a discrete probability measure on* $\mathcal{S}^{n \times n} \times \mathbb{R}_+ \times \mathbb{R}_+$ *by setting* $\omega_i = (H_i, L_i, p_i)$ *and* $\mathbb{P}(\omega = \omega_i) = p_i$ *for* $i = 1, ..., m$. *Hence, in the context of Theorem C.3.1,* $G = \mathbb{E}[\frac{1}{\gamma} H] = \sum_{i=1}^m H_i = \text{blkdiag}(B_1^{-1}, B_2^{-1}, ..., B_m^{-1})$. *We first verify the gradient inequality (C.16). For*

Algorithm 2 Accelerated randomized coordinate descent.

Require: $\mu > 0$, partition $\{J_i\}_{i=1}^m$, positive definite $\{B_i\}_{i=1}^m$, Lipschitz constants $\{L_i\}_{i=1}^m$, $x_0 \in \mathbb{R}^n$.

Set $\tau = \frac{\sqrt{\mu}}{\sum_{i=1}^m \sqrt{L_i}}$.

Set $H_i = S_i B_i^{-1} S_i^\mathsf{T}$ for $i = 1, ..., m$. `// ` S_i `denotes the column selector for partition` J_i`.`

Set $p_i = \frac{\sqrt{L_i}}{\sum_{j=1}^m \sqrt{L_j}}$ for $i = 1, ..., m$.

Set $y_0 = z_0 = x_0$.

for $k = 0, ..., T-1$ **do**

  $i_k \leftarrow$ random sample from $\{1, ..., m\}$ with $\mathbb{P}(i_k = i) = p_i$.

  $x_{k+1} = \frac{1}{1+\tau} y_k + \frac{\tau}{1+\tau} z_k$.

  $y_{k+1} = x_{k+1} - \frac{1}{L_{i_k}} H_{i_k} \nabla f(x_{k+1})$.

  $z_{k+1} = z_k + \tau(x_{k+1} - z_k) - \frac{\tau}{\mu p_{i_k}} H_{i_k} \nabla f(x_{k+1})$.

end for

Return y_T.

every fixed $x \in \mathbb{R}^n$, for every $i = 1, ..., m$ there exists a $c_i \in \mathbb{R}^n$ such that

$$\begin{aligned} f(\Phi(x;\omega_i)) &= f(x) - \frac{1}{L_i}\langle \nabla f(x), H_i \nabla f(x)\rangle + \frac{1}{2L_i^2} \nabla f(x)^\mathsf{T} H_i \nabla^2 f(c_i) H_i \nabla f(x) \\ &= f(x) - \frac{1}{L_i}\langle \nabla f(x), H_i \nabla f(x)\rangle \\ &\quad + \frac{1}{2L_i^2} \nabla f(x)^\mathsf{T} S_i B_i^{-1/2} B_i^{-1/2} S_i^\mathsf{T} \nabla^2 f(c_i) S_i B_i^{-1/2} B_i^{-1/2} S_i^\mathsf{T} \nabla f(x) \\ &\leq f(x) - \frac{1}{L_i}\langle \nabla f(x), H_i \nabla f(x)\rangle + \frac{1}{2L_i} \nabla f(x)^\mathsf{T} S_i B_i^{-1} S_i^\mathsf{T} \nabla f(x) \\ &= f(x) - \frac{1}{2L_i} \|\nabla f(x)\|_{H_i}^2 . \end{aligned}$$

We next compute the ν constant defined in (C.17). We do this by checking the sufficient condition that $H_i G^{-1} H_i \preccurlyeq \nu H_i$ for $i = 1, ..., m$. Doing so yields that $\nu = 1$, since

$$H_i G^{-1} H_i = S_i B_i^{-1} S_i^\mathsf{T} \mathrm{blkdiag}(B_1, B_2, ..., B_m) S_i B_i^{-1} S_i^\mathsf{T} = S_i B_i^{-1} B_i B_i^{-1} S_i^\mathsf{T} = S_i B_i^{-1} S_i^\mathsf{T} = H_i .$$

To complete the argument, we set μ as the strong convexity constant and L as the Lipschitz gradient constant of f with respect to the $\|\cdot\|_{G^{-1}}$ norm. It is straightforward to check that

$$\mu = \inf_{x \in \mathbb{R}^n} \lambda_{\max}(G^{1/2} \nabla^2 f(x) G^{1/2}) , \quad L = \sup_{x \in \mathbb{R}^n} \lambda_{\max}(G^{1/2} \nabla^2 f(x) G^{1/2}) .$$

We now argue that $\sqrt{L} \leq \sum_{i=1}^m \sqrt{L_i}$. *Let* $x \in \mathbb{R}^n$ *achieve the supremum in the definition of* L *(if no such* x *exists, then let* x *be arbitrarily close and take limits). Then,*

$$\begin{aligned}
L &= \lambda_{\max}(G^{1/2}\nabla^2 f(x) G^{1/2}) = \lambda_{\max}((\nabla^2 f(x))^{1/2} G (\nabla^2 f(x))^{1/2}) \\
&= \lambda_{\max}\left((\nabla^2 f(x))^{1/2}\left(\sum_{i=1}^m S_i B_i^{-1} S_i^\mathsf{T}\right)(\nabla^2 f(x))^{1/2}\right) \\
&\overset{(a)}{\leq} \sum_{i=1}^m \lambda_{\max}((\nabla^2 f(x))^{1/2} S_i B_i^{-1} S_i^\mathsf{T} (\nabla^2 f(x))^{1/2}) \\
&\overset{(b)}{=} \sum_{i=1}^m \lambda_{\max}(S_i S_i^\mathsf{T} \nabla^2 f(x) S_i S_i^\mathsf{T} S_i B_i^{-1} S_i^\mathsf{T}) \\
&= \sum_{i=1}^m \lambda_{\max}((S_i B_i^{-1} S_i^\mathsf{T})^{1/2} S_i S_i^\mathsf{T} \nabla^2 f(x) S_i S_i^\mathsf{T} (S_i B_i^{-1} S_i^\mathsf{T})^{1/2}) \\
&\overset{(c)}{=} \sum_{i=1}^m \lambda_{\max}(S_i B_i^{-1/2} S_i^\mathsf{T} S_i S_i^\mathsf{T} \nabla^2 f(x) S_i S_i^\mathsf{T} S_i B_i^{-1/2} S_i^\mathsf{T}) \\
&\overset{(d)}{=} \sum_{i=1}^m \lambda_{\max}(B_i^{-1/2} S_i^\mathsf{T} \nabla^2 f(x) S_i B_i^{-1/2}) \leq \sum_{i=1}^m L_i \,.
\end{aligned}$$

Above, (a) follows by the convexity of the maximum eigenvalue, (b) holds since $S_i^\mathsf{T} S_i = I$, *(c) uses the fact that for any matrix* Q *satisfying* $Q^\mathsf{T} Q = I$ *and* M *positive semi-definite, we have* $(QMQ^\mathsf{T})^{1/2} = QM^{1/2}Q^\mathsf{T}$, *and (d) follows since* $\lambda_{\max}(S_i M S_i^\mathsf{T}) = \lambda_{\max}(M)$ *for any* $p \times p$ *symmetric matrix* M. *Using the fact that* $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ *for any non-negative* a, b, *the inequality* $\sqrt{L} \leq \sum_{i=1}^m \sqrt{L_i}$ *immediately follows. To conclude the proof, it remains to calculate the requirement on* τ *via* (C.18). *Since* $\frac{\gamma_i}{\sqrt{\Gamma_i}} = \frac{p_i}{\sqrt{L_i}} = \frac{1}{\sum_{i=1}^m \sqrt{L_i}}$, *we have that* $\frac{\gamma_i}{\sqrt{\Gamma_i}} \leq \frac{1}{\sqrt{L}}$, *and hence the requirement is that* $\tau \leq \frac{\sqrt{\mu}}{\sum_{i=1}^m \sqrt{L_i}}$.

C.4.2 Relating Algorithm 2 to ACDM

For completeness, we replicate the description of the ACDM algorithm from Nesterov and Stich in Algorithm 3. We make one minor tweak in the initialization of the A_k, B_k *sequence which greatly simplifies the exposition of what follows.*

We first write the sequence produced by Algorithm 3 as

$$y_k = \frac{(1-\alpha_k)x_k + \alpha_k(1-\beta_k)z_k}{1-\alpha_k\beta_k}\,, \tag{C.26a}$$

$$x_{k+1} = y_k - \frac{1}{L_{i_k}} H_{i_k} \nabla f(y_k)\,, \tag{C.26b}$$

$$z_{k+1} - z_k = \beta_k \left(y_k - z_k - \frac{a_{k+1}}{B_{k+1} p_{i_k} \beta_k} H_{i_k} \nabla f(y_k)\right)\,. \tag{C.26c}$$

Algorithm 3 ACDM from Nesterov and Stich [48], $\alpha = 1, \beta = 1/2$ case.

Require: $\mu > 0$, partition $\{J_i\}_{i=1}^m$, positive definite $\{B_i\}_{i=1}^m$, Lipschitz constants $\{L_i\}_{i=1}^m$, $x_0 \in \mathbb{R}^n$.

Set $H_i = S_i B_i^{-1} S_i^\mathsf{T}$ for $i = 1, ..., m$. `// ` S_i `denotes the column selector for partition` J_i`.`
Set $p_i = \frac{\sqrt{L_i}}{\sum_{j=1}^m \sqrt{L_j}}$ for $i = 1, ..., m$.
Set $A_0 = 1, B_0 = \mu$. `// Modified from` $A_0 = 0, B_0 = 1$`.`
Set $S_{1/2} = \sum_{i=1}^m \sqrt{L_i}$.
Set $y_0 = z_0 = x_0$.
for $k = 0, ..., T-1$ **do**
  $i_k \leftarrow$ random sample from $\{1, ..., m\}$ with $\mathbb{P}(i_k = i) = p_i$.
  $a_{k+1} \leftarrow$ positive solution to $a_{k+1}^2 S_{1/2}^2 = (A_k + a_{k+1})(B_k + \mu a_{k+1})$.
  $A_{k+1} = A_k + a_{k+1}, B_{k+1} = B_k + \mu a_{k+1}$.
  $\alpha_k = \frac{a_{k+1}}{A_{k+1}}, \beta_k = \mu \frac{a_{k+1}}{B_{k+1}}$.
  $y_k = \frac{(1-\alpha_k)x_k + \alpha_k(1-\beta_k)z_k}{1-\alpha_k\beta_k}$.
  $x_{k+1} = y_k - \frac{1}{L_{i_k}} H_{i_k} \nabla f(y_k)$.
  $z_{k+1} = (1-\beta_k)z_k + \beta_k y_k - \frac{a_{k+1}}{B_{k+1} p_{i_k}} H_{i_k} \nabla f(y_k)$.
end for
Return x_T.

Since $\beta_k B_{k+1} = \mu a_{k+1}$, the z_{k+1} update simplifies to

$$z_{k+1} - z_k = \beta_k \left(y_k - z_k - \frac{1}{\mu p_{i_k}} H_{i_k} \nabla f(y_k) \right) .$$

A simple calculation shows that

$$(1 - \alpha_k\beta_k) y_k = (1-\alpha_k) x_k + \alpha_k (1-\beta_k) z_k ,$$

from which we conclude that

$$\frac{\alpha_k(1-\beta_k)}{1-\alpha_k}(y_k - z_k) = x_k - y_k . \tag{C.27}$$

Observe that

$$A_{k+1} = \sum_{i=1}^{k+1} a_i + A_0 , \quad B_{k+1} = \mu \sum_{i=1}^{k+1} a_i + B_0 .$$

Hence as long as $\mu A_0 = B_0$ (which is satisfied by our modification), we have that $\mu A_{k+1} = B_{k+1}$ for all $k \geq 0$. With this identity, we have that $\alpha_k = \beta_k$ for all $k \geq 0$. Therefore, (C.27) simplifies to

$$\beta_k (y_k - z_k) = x_k - y_k .$$

We now calculate the value of β_k. At every iteration, we have that

$$a_{k+1}^2 S_{1/2}^2 = A_{k+1}B_{k+1} = \frac{1}{\mu}B_{k+1}^2 \Longrightarrow \frac{a_{k+1}}{B_{k+1}} = \frac{1}{\sqrt{\mu}S_{1/2}} .$$

By the definition of β_k,

$$\beta_k = \mu\frac{a_{k+1}}{B_{k+1}} = \frac{\sqrt{\mu}}{S_{1/2}} = \frac{\sqrt{\mu}}{\sum_{i=1}^m \sqrt{L_i}} = \tau .$$

Combining these identities, we have shown that (C.26a), (C.26b), and (C.26c) simplifies to

$$y_k = \frac{1}{1+\tau}x_k + \frac{\tau}{1+\tau}z_k , \tag{C.28a}$$

$$x_{k+1} = y_k - \frac{1}{L_{i_k}}H_{i_k}\nabla f(y_k) , \tag{C.28b}$$

$$z_{k+1} - z_k = \tau\left(y_k - z_k - \frac{1}{\mu p_{i_k}}H_{i_k}\nabla f(y_k)\right) . \tag{C.28c}$$

This sequence directly coincides with the sequence generated by Algorithm 2 after a simple relabeling.

C.4.3 Accelerated Gauss-Seidel for fixed partitions from ACDM

Algorithm 4 Accelerated randomized block Gauss-Seidel for fixed partitions [48].

Require: $A \in \mathbb{R}^{n\times n}$, $A \succ 0$, $b \in \mathbb{R}^n$, $x_0 \in \mathbb{R}^n$, block size p, μ_{part} defined in (3.21).

Set $A_0 = 0, B_0 = 1$.
Set $\sigma = \frac{n}{p}\mu_{\text{part}}$.
Set $y_0 = z_0 = x_0$.
for $k = 0, ..., T-1$ **do**
  $i_k \leftarrow$ uniform from $\{1, 2, ..., n/p\}$.
  $S_k \leftarrow$ column selector associated with partition J_{i_k}.
  $a_{k+1} \leftarrow$ positive solution to $a_{k+1}^2(n/p)^2 = (A_k + a_{k+1})(B_k + \sigma a_{k+1})$.
  $A_{k+1} = A_k + a_{k+1}, B_{k+1} = B_k + \sigma a_{k+1}$.
  $\alpha_k = \frac{a_{k+1}}{A_{k+1}}, \beta_k = \sigma\frac{a_{k+1}}{B_{k+1}}$.
  $y_k = \frac{(1-\alpha_k)x_k + \alpha_k(1-\beta_k)z_k}{1-\alpha_k\beta_k}$.
  $x_{k+1} = y_k - S_k(S_k^\mathsf{T}AS_k)^{-1}S_k^\mathsf{T}(Ay_k - b)$.
  $z_{k+1} = (1-\beta_k)z_k + \beta_k y_k - \frac{na_{k+1}}{pB_{k+1}}S_k(S_k^\mathsf{T}AS_k)^{-1}S_k^\mathsf{T}(Ay_k - b)$.
end for
Return x_T.

We now describe Algorithm 4, which is the specialization of ACDM (Algorithm 3) to accelerated Gauss-Seidel in the fixed partition setting.

As mentioned previously, we set the function $f(x) = \frac{1}{2}x^\mathsf{T}Ax - x^\mathsf{T}b$. *Given a partition* $\{J_i\}_{i=1}^{n/p}$, *we let* $B_i = S_i^\mathsf{T}AS_i$, *where* $S_i \in \mathbb{R}^{n\times p}$ *is the column selector matrix associated to the partition* J_i. *With this setting, we have that* $L_1 = L_2 = ... = L_{n/p} = 1$, *and hence we have* $p_i = p/n$ *for all* i *(i.e. the sampling distribution is uniform over all partitions). We now need to compute the strong convexity constant* μ. *With the simplifying assumption that the partitions are ordered,* μ *is simply the strong convexity constant with respect to the norm induced by the matrix* $\text{blkdiag}(B_1, B_2, ..., B_{n/p})$. *Hence, using the definition of* μ_{part} *from (3.21), we have that* $\mu = \frac{n}{p}\mu_{\text{part}}$. *Algorithm 4 now follows from plugging our particular choices of* f *and the constants into Algorithm 3.*

C.5 A Result for Randomized Block Kaczmarz

We now use Theorem C.3.1 to derive a result similar to Theorem 3.4.5 for the randomized accelerated Kaczmarz algorithm. In this setting, we let $A \in \mathbb{R}^{m\times n}$, $m \geq n$ *be a matrix with full column rank, and* $b \in \mathbb{R}^m$ *such that* $b \in \mathcal{R}(A)$. *That is, there exists a unique* $x_* \in \mathbb{R}^n$ *such that* $Ax_* = b$. *We note that this section generalizes the result of [33] to the block case (although the proof strategy is quite different).*

We first describe the randomized accelerated block Kaczmarz algorithm in Algorithm 5. Our main convergence result concerning Algorithm 5 is presented in Theorem C.5.1.

Algorithm 5 Accelerated randomized block Kaczmarz.

Require: $A \in \mathbb{R}^{m\times n}$, A full column rank, $b \in \mathcal{R}(A)$, sketching matrices $\{S_k\}_{k=0}^{T-1} \subseteq \mathbb{R}^{m\times p}$, $x_0 \in \mathbb{R}^n$, $\mu \in (0,1)$, $\nu \geq 1$.

Set $\tau = \sqrt{\mu/\nu}$.
Set $y_0 = z_0 = x_0$.
for $k = 0, ..., T-1$ **do**
  $x_{k+1} = \frac{1}{1+\tau}y_k + \frac{\tau}{1+\tau}z_k$.
  $y_{k+1} = x_{k+1} - (S_k^\mathsf{T}A)^\dagger S_k^\mathsf{T}(Ax_{k+1} - b)$.
  $z_{k+1} = z_k + \tau(x_{k+1} - z_k) - \frac{\tau}{\mu}(S_k^\mathsf{T}A)^\dagger S_k^\mathsf{T}(Ax_{k+1} - b)$.
end for
Return y_T.

Theorem C.5.1. *(Theorem 3.4.7 restated.) Let* A *be an* $m \times n$ *matrix with full column rank, and* $b \in \mathcal{R}(A)$. *Let* $x_* \in \mathbb{R}^n$ *denote the unique vector satisfying* $Ax_* = b$. *Suppose each* S_k, $k = 0, 1, 2, ...$ *is an independent copy of a random sketching matrix* $S \in \mathbb{R}^{m\times p}$. *Let* $\mu = \lambda_{\min}(\mathbb{E}[P_{A^\mathsf{T}S}])$. *Suppose the distribution of* S *satisfies* $\mu > 0$. *Invoke Algorithm 5 with* μ *and* ν, *where* ν *is defined as*

$$\nu = \lambda_{\max}\left(\mathbb{E}\left[(G^{-1/2}HG^{-1/2})^2\right]\right), \quad G = \mathbb{E}[H], \quad H = P_{A^\mathsf{T}S}. \tag{C.29}$$

Then for all $k \geq 0$ we have

$$\mathbb{E}[\|y_k - x_*\|_2] \leq \sqrt{2}\left(1 - \sqrt{\frac{\mu}{\nu}}\right)^{k/2} \|x_0 - x_*\|_2 \,. \tag{C.30}$$

Proof. The proof is very similar to that of Theorem 3.4.5, so we only sketch the main argument. The key idea is to use the correspondence between randomized Kaczmarz and coordinate descent (see e.g. Section 5.2 of [29]). To do this, we apply Theorem C.3.1 to $f(x) = \frac{1}{2}\|x - x_*\|_2^2$. As in the proof of Theorem 3.4.5, we construct a probability measure on $\mathcal{S}^{n\times n} \times \mathbb{R}_+ \times \mathbb{R}_+$ from the given random matrix H by considering the random variable $(H, 1, 1)$. To see that the sequence (C.14a), (C.14b), and (C.14c) induces the same update sequence as Algorithm 5, the crucial step is to notice that

$$\begin{aligned} H_k \nabla f(x_{k+1}) &= P_{A^\mathsf{T} S_k} \nabla f(x_{k+1}) = A^\mathsf{T} S_k (S_k^\mathsf{T} A A^\mathsf{T} S_k)^\dagger S_k^\mathsf{T} A (x_{k+1} - x_*) \\ &= A^\mathsf{T} S_k (S_k^\mathsf{T} A A^\mathsf{T} S_k)^\dagger S_k^\mathsf{T} (A x_{k+1} - b) = (S_k^\mathsf{T} A)^\dagger S_k^\mathsf{T} (A x_{k+1} - b) \,. \end{aligned}$$

Next, the fact that f is $\lambda_{\min}(\mathbb{E}[P_{A^\mathsf{T} S}])$-strongly convex and 1-Lipschitz with respect to the $\|\cdot\|_{G^{-1}}$ norm, where $G = \mathbb{E}[P_{A^\mathsf{T} S}]$, follows immediately by a nearly identical argument used in the proof of Theorem 3.4.5. It remains to check the gradient inequality (C.16). Let $x \in \mathbb{R}^n$ be fixed. Then using the fact that f is quadratic, for almost every $\omega \in \Omega$,

$$\begin{aligned} f(\Phi(x;\omega)) &= f(x) - \langle \nabla f(x), H(x - x_*)\rangle + \frac{1}{2}\|H(x - x_*)\|_2^2 \\ &= f(x) - \langle x - x_*, P_{A^\mathsf{T} S}(x - x_*)\rangle + \frac{1}{2}\|P_{A^\mathsf{T} S}(x - x_*)\|_2^2 \\ &= f(x) - \frac{1}{2}\langle x - x_*, P_{A^\mathsf{T} S}(x - x_*)\rangle \,. \end{aligned}$$

Hence the gradient inequality (C.16) holds with equality. □

C.5.1 Computing ν and μ in the setting of [33]

We first state a proposition which will be useful in our analysis of ν.

Proposition C.5.2. *Let $M_1, \ldots, M_s \subseteq \mathbb{R}^n$ denote subspaces of $\mathbb{R}^n$ such that $M_1 + \ldots + M_s = \mathbb{R}^n$. Then we have*

$$\sum_{i=1}^s P_{M_i} \left(\sum_{i=1}^s P_{M_i}\right)^{-1} P_{M_i} \preccurlyeq \sum_{i=1}^s P_{M_i} \,.$$

Proof. We will prove that for every $1 \leq i \leq s$,

$$P_{M_i} \left(\sum_{i=1}^s P_{M_i}\right)^{-1} P_{M_i} \preccurlyeq P_{M_i} \,, \tag{C.31}$$

from which the claim immediately follows. By Schur complements, (C.31) holds iff

$$0 \preccurlyeq \begin{bmatrix} P_{M_i} & P_{M_i} \\ P_{M_i} & \sum_{i=1}^{s} P_{M_i} \end{bmatrix} = \begin{bmatrix} P_{M_i} & P_{M_i} \\ P_{M_i} & P_{M_i} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \sum_{j\neq i}^{s} P_{M_j} \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \otimes P_{M_i} + \begin{bmatrix} 0 & 0 \\ 0 & \sum_{j\neq i}^{s} P_{M_j} \end{bmatrix} .$$

Since the eigenvalues of a Kronecker product are given by the Cartesian product of the individual eigenvalues, (C.31) holds. □

Now we can estimate the ν and μ values. Let $a_i \in \mathbb{R}^n$ denote each row of A, with $\|a_i\|_2 = 1$ for all $i = 1, ..., m$. In this setting, $H = P_{a_i} = a_i a_i^\mathsf{T}$ with probability $1/m$. Hence, $G = \mathbb{E}[H] = \sum_{i=1}^{m} \frac{1}{m} a_i a_i^\mathsf{T} = \frac{1}{m} A^\mathsf{T} A$. Furthermore,

$$\mathbb{E}[HG^{-1}H] = \sum_{i=1}^{m} a_i a_i^\mathsf{T} m (A^\mathsf{T}A)^{-1} a_i a_i^\mathsf{T} \frac{1}{m}$$
$$= \sum_{i=1}^{m} a_i a_i^\mathsf{T} (A^\mathsf{T}A)^{-1} a_i a_i^\mathsf{T}$$
$$\overset{(a)}{\preccurlyeq} \sum_{i=1}^{m} a_i a_i^\mathsf{T} = A^\mathsf{T}A = mG ,$$

where (a) follows from Proposition C.5.2. Hence, $\nu \neq m$. On the other hand,

$$\mu = \lambda_{\min}(\mathbb{E}[P_{A^\mathsf{T}S}]) = \lambda_{\min}(G) = \frac{1}{m}\lambda_{\min}(A^\mathsf{T}A) .$$

C.6 Proofs for Random Coordinate Sampling (Section 3.4.3.3)

Our primary goal in this section is to provide a proof of Lemma 3.4.8. Along the way, we prove a few other results which are of independent interest. We first provide a proof of the lower bound claim in Lemma 3.4.8.

Proposition C.6.1. *Let A be an $n \times n$ matrix and let $S \in \mathbb{R}^{n\times p}$ be a random matrix. Put $G = \mathbb{E}[P_{A^{1/2}S}]$ and suppose that G is positive definite. Let $\nu > 0$ be any positive number such that*

$$\mathbb{E}[P_{A^{1/2}S} G^{-1} P_{A^{1/2}S}] \preccurlyeq \nu G , \quad G = \mathbb{E}[P_{A^{1/2}S}] . \tag{C.32}$$

Then $\nu \geq n/p$.

Proof. Since trace commutes with expectation and respects the positive semi-definite ordering, taking trace of both sides of (C.32) yields that

$$\begin{aligned} n &= \mathbf{Tr}(GG^{-1}) = \mathbf{Tr}(\mathbb{E}[P_{A^{1/2}S}G^{-1}]) = \mathbb{E}[\mathbf{Tr}(P_{A^{1/2}S}G^{-1})] = \mathbb{E}[\mathbf{Tr}(P_{A^{1/2}S}G^{-1}P_{A^{1/2}S})] \\ &= \mathbf{Tr}(\mathbb{E}[P_{A^{1/2}S}G^{-1}P_{A^{1/2}S}]) \overset{(C.32)}{\leq} \nu\mathbf{Tr}(\mathbb{E}[P_{A^{1/2}S}]) \\ &= \nu\mathbb{E}[\mathbf{Tr}(P_{A^{1/2}S})] = \nu\mathbb{E}[\mathrm{rank}(A^{1/2}S)] \leq \nu p\,. \end{aligned}$$

□

Next, the upper bound relies on the following lemma, which generalizes Lemma 2 of [57].

Lemma C.6.2. *Let M be a random matrix. We have that*

$$\mathbb{E}[P_M] \succcurlyeq \mathbb{E}[M](\mathbb{E}[M^\mathsf{T}M])^\dagger\mathbb{E}[M^\mathsf{T}]\,. \tag{C.33}$$

Proof. Our proof follows the strategy in the proof of Theorem 3.2 from [78]. First, write $P_B = B(B^\mathsf{T}B)^\dagger B^\mathsf{T}$. Since $\mathcal{R}(B^\mathsf{T}) = \mathcal{R}(B^\mathsf{T}B)$, we have by generalized Schur complements (see e.g. Theorem 1.20 from [78]) and the fact that expectation preserves the semi-definite order,

$$\begin{bmatrix} B^\mathsf{T}B & B^\mathsf{T} \\ B & P_B \end{bmatrix} \succcurlyeq 0 \Longrightarrow \begin{bmatrix} \mathbb{E}[B^\mathsf{T}B] & \mathbb{E}[B^\mathsf{T}] \\ \mathbb{E}[B] & \mathbb{E}[P_B] \end{bmatrix} \succcurlyeq 0\,.$$

To finish the proof, we need to argue that $\mathcal{R}(\mathbb{E}[B^\mathsf{T}]) \subseteq \mathcal{R}(\mathbb{E}[B^\mathsf{T}B])$, which would allow us to apply the generalized Schur complement again to the right hand side. Fix a $z \in \mathcal{R}(\mathbb{E}[B^\mathsf{T}])$; we can write $z = \mathbb{E}[B^\mathsf{T}]y$ for some y. Now let $q \in \mathrm{Kern}(\mathbb{E}[B^\mathsf{T}B])$. We have that $\mathbb{E}[B^\mathsf{T}B]q = 0$, which implies $0 = q^\mathsf{T}\mathbb{E}[B^\mathsf{T}B]q = \mathbb{E}[\|Bq\|_2^2]$. Therefore, $Bq = 0$ a.s. But this means that $z^\mathsf{T}q = \mathbb{E}[y^\mathsf{T}Bq] = 0$. Hence, $z \in \mathrm{Kern}(\mathbb{E}[B^\mathsf{T}B])^\perp = \mathcal{R}(\mathbb{E}[B^\mathsf{T}B])$. Now applying the generalized Schur complement one more time yields the claim. □

We are now in a position to prove the upper bound of Lemma 3.4.8. We apply Lemma C.6.2 to $M = A^{1/2}SS^\mathsf{T}A^{1/2}$ to conclude, using the fact that $\mathcal{R}(M) = \mathcal{R}(MM^\mathsf{T})$, that

$$\mathbb{E}[P_{A^{1/2}S}] = \mathbb{E}[P_{A^{1/2}SS^\mathsf{T}A^{1/2}}] \succcurlyeq \mathbb{E}[A^{1/2}SS^\mathsf{T}A^{1/2}](\mathbb{E}[A^{1/2}SS^\mathsf{T}ASS^\mathsf{T}A^{1/2}])^\dagger\mathbb{E}[A^{1/2}SS^\mathsf{T}A^{1/2}]\,. \tag{C.34}$$

Elementary calculations now yield that for any fixed symmetric matrix $A \in \mathbb{R}^{n\times n}$,

$$\mathbb{E}[SS^\mathsf{T}] = \frac{p}{n}I, \qquad \mathbb{E}[SS^\mathsf{T}ASS^\mathsf{T}] = \frac{p}{n}\left(\frac{p-1}{n-1}A + \left(1 - \frac{p-1}{n-1}\right)\mathrm{diag}(A)\right)\,. \tag{C.35}$$

Hence plugging (C.35) into (C.34),

$$\mathbb{E}[P_{A^{1/2}S}] \succcurlyeq \frac{p}{n}\left(\frac{p-1}{n-1}I + \left(1 - \frac{p-1}{n-1}\right)A^{-1/2}\mathrm{diag}(A)A^{-1/2}\right)^{-1}\,. \tag{C.36}$$

We note that the lower bound (3.23) *for* μ_{rand} *presented in Section 3.4.2 follows immediately from* (C.36).

We next manipulate (3.31) *in order to use* (C.36). *Recall that* $G = \mathbb{E}[H]$ *and* $H = S(S^\mathsf{T}AS)^\dagger S^\mathsf{T}$. *From* (C.24), *we have*

$$\lambda_{\max}\left(\mathbb{E}\left[(G^{-1/2}HG^{-1/2})^2\right]\right) \le \nu \iff \mathbb{E}\left[HG^{-1}H\right] \preccurlyeq \nu G\,.$$

Next, a simple computation yields

$$\mathbb{E}[HG^{-1}H] = \mathbb{E}[S(S^\mathsf{T}AS)^{-1}S^\mathsf{T}G^{-1}S(S^\mathsf{T}AS)^{-1}S^\mathsf{T}] = A^{-1/2}\mathbb{E}[P_{A^{1/2}S}(\mathbb{E}[P_{A^{1/2}S}])^{-1}P_{A^{1/2}S}]A^{-1/2}\,.$$

Again, since conjugation by $A^{1/2}$ *preserves semi-definite ordering, we have that*

$$\mathbb{E}[HG^{-1}H] \preccurlyeq \nu G \iff \mathbb{E}[P_{A^{1/2}S}(\mathbb{E}[P_{A^{1/2}S}])^{-1}P_{A^{1/2}S}] \preccurlyeq \nu\mathbb{E}[P_{A^{1/2}S}]\,.$$

Using the fact that for positive definite matrices X, Y *we have* $X \preccurlyeq Y$ *iff* $Y^{-1} \preccurlyeq X^{-1}$, (C.36) *is equivalent to*

$$(\mathbb{E}[P_{A^{1/2}S}])^{-1} \preceq \frac{n}{p}\left(\frac{p-1}{n-1}I + \left(1 - \frac{p-1}{n-1}\right)A^{-1/2}\mathrm{diag}(A)A^{-1/2}\right)\,.$$

Conjugating both sides by $P_{A^{1/2}S}$ *and taking expectations,*

$$\mathbb{E}[P_{A^{1/2}S}(\mathbb{E}[P_{A^{1/2}S}])^{-1}P_{A^{1/2}S}] \preceq \frac{n}{p}\left(\frac{p-1}{n-1}\mathbb{E}[P_{A^{1/2}S}] + \left(1 - \frac{p-1}{n-1}\right)\mathbb{E}[P_{A^{1/2}S}A^{-1/2}\mathrm{diag}(A)A^{-1/2}P_{A^{1/2}S}]\right)\,. \tag{C.37}$$

Next, letting $J \subseteq 2^{[n]}$ *denote the index set associated to* S, *for every* S *we have*

$$\begin{aligned}
&P_{A^{1/2}S}A^{-1/2}\mathrm{diag}(A)A^{-1/2}P_{A^{1/2}S}\\
&\quad= A^{1/2}S(S^\mathsf{T}AS)^{-1}S^\mathsf{T}A^{1/2}A^{-1/2}\mathrm{diag}(A)A^{-1/2}A^{1/2}S(S^\mathsf{T}AS)^{-1}S^\mathsf{T}A^{1/2}\\
&\quad= A^{1/2}S(S^\mathsf{T}AS)^{-1/2}(S^\mathsf{T}AS)^{-1/2}(S^\mathsf{T}\mathrm{diag}(A)S)(S^\mathsf{T}AS)^{-1/2}(S^\mathsf{T}AS)^{-1/2}S^\mathsf{T}A^{1/2}\\
&\quad\preccurlyeq \lambda_{\max}((S^\mathsf{T}\mathrm{diag}(A)S)(S^\mathsf{T}AS)^{-1})A^{1/2}S(S^\mathsf{T}AS)^{-1}S^\mathsf{T}A^{1/2}\\
&\quad\preccurlyeq \frac{\max_{i\in J}A_{ii}}{\lambda_{\min}(A_J)}P_{A^{1/2}S}\\
&\quad\preccurlyeq \max_{J\in 2^{[n]}:|J|=p}\kappa_{\text{eff},J}(A)P_{A^{1/2}S}\,.
\end{aligned}$$

Plugging this calculation back into (C.37) *yields the desired upper bound of Lemma 3.4.8.*

Bibliography

[1] Zeyuan Allen-Zhu, Peter Richtárik, Zheng Qu, and Yang Yuan. "Even Faster Accelerated Coordinate Descent Using Non-Uniform Sampling". In: *ICML*. 2016.

[2] Felipe Alvarez, Jérôme Bolte, and Olivier Brahic. In: *SIAM Journal on Control and Optimization* 43.2 (2004), pp. 477–501.

[3] Shun-Ichi Amari. "Natural Gradient Works Efficiently in Learning". In: *Neural Computation* (1998), pp. 251–276.

[4] Michael Jordan Ashia Wilson Benjamin Recht. *A Lyapunov Analysis of Momentum Methods in Optimization*. Arxiv preprint arXiv1611.02635. 2016.

[5] Haim Avron, Kenneth L. Clarkson, and David P. Woodruff. "Faster Kernel Ridge Regression Using Sketching and Preconditioning". In: *arXiv* 1611.03220 (2017).

[6] Michel Baes. *Estimate sequence methods: Extensions and approximations*. Aug. 2009.

[7] John C. Baez and Blake S. Pollard. "Relative Entropy in Biological Systems". In: *Entropy* 18.2 (2016), p. 46.

[8] Amir Beck and Marc Teboulle. "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems". In: *SIAM Journal on Imaging Sciences* 2.1 (2009), pp. 183–202.

[9] Sébastien Bubeck, Yin Tat Lee, and Mohit Singh. "A geometric alternative to Nesterov's accelerated gradient descent". In: *ArXiv preprint arXiv:1506.08187* (2015).

[10] P. L Chebyshev. "Théorie des mécanismes connus sous le nom de parallélogrammes". In: *Mémoires Présentés á l'Académie Impériale des Sciences de St-Pétersbourg* VII.539-568 (1854).

[11] Gong Chen and Marc Teboulle. "Convergence Analysis of a Proximal-Like Minimization Algorithm Using Bregman Functions". In: 3 (Aug. 1993).

[12] Adam Coates and Andrew Y. Ng. "Learning Feature Representations with K-Means". In: *Neural Networks: Tricks of the Trade*. Springer, 2012.

[13] Jelena Diakonikolas and Lorenzo Orecchia. "Accelerated Extra-Gradient Descent: A Novel Accelerated First-Order Method". In: *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*. 2018, 23:1–23:19.

[14] Yoel Drori and Marc Teboulle. "Performance of first-order methods for smooth convex minimization: a novel approach". In: *Math. Program.* 145.1-2 (2014), pp. 451–482.

[15] Dmitry Drusvyatskiy, Maryam Fazel, and Scott Roy. "An optimal first order method based on optimal quadratic averaging". In: *ArXiv preprint arXiv:1604.06543* (2016).

[16] John Duchi, Elad Hazan, and Yoram Singer. *Adaptive Subgradient Methods for Online Learning and Stochastic Optimization*. Tech. rep. EECS Department, University of California, Berkeley, 2010.

[17] Olivier Fercoq and Peter Richtárik. "Accelerated, Parallel, and Proximal Coordinate Descent". In: *SIAM J. Optim.* 25.4 (2015).

[18] Kimon Fountoulakis and Rachael Tappenden. "A Flexible Coordinate Descent Method". In: *arXiv* 1507.03713 (2016).

[19] Peter Giesl and Sigurdur F. Hafstein. "Construction of Lyapunov functions for nonlinear planar systems by linear programming". In: 2011.

[20] Robert M. Gower and Peter Richtárik. "Randomized Iterative Methods for Linear Systems". In: *SIAM Journal on Matrix Analysis and Applications* 36 (4 2015).

[21] Geovani Nunes Grapiglia and Yurii Nesterov. "Regularized Newton Methods for Minimizing Functions with Hölder Continuous Hessians". In: *SIAM Journal on Optimization* 27.1 (2017), pp. 478–506.

[22] Marc Harper. "The Replicator Equation as an Inference Dynamic". In: (Nov. 2009).

[23] "Information Theory and Statistical Mechanics". In: *Phys. Rev.* 106.620-630 (1957).

[24] Sahar Karimi and Stephen Vavasis. "A unified convergence bound for conjugate gradient and accelerated gradient". In: (May 2016).

[25] Walid Krichene, Alexandre Bayen, and Peter Bartlett. "Accelerated Mirror Descent in Continuous and Discrete Time". In: *Advances in Neural Information Processing Systems (NIPS) 29*. 2015.

[26] Walid Krichene, Alexandre Bayen, and Peter L Bartlett. "Accelerated Mirror Descent in Continuous and Discrete Time". In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, Inc., 2015, pp. 2845–2853.

[27] Joseph P LaSalle and Solomon Lefschetz. *Stability by Liapunov's Direct Method, with Applications*. Academic Press, 1961.

[28] Ching-Pei Lee and Stephen J. Wright. "Random Permutations Fix a Worst Case for Cyclic Coordinate Descent". In: *arXiv* 1607.08320 (2016).

[29] Yin Tat Lee and Aaron Sidford. "Efficient Accelerated Coordinate Descent Methods and Faster Algorithms for Solving Linear Systems". In: *FOCS*. 2013.

[30] Laurent Lessard, Benjamin Recht, and Andrew Packard. "Analysis and Design of Optimization Algorithms via Integral Quadratic Constraints". In: *SIAM Journal on Optimization* 26.1 (), pp. 57–95.

[31] Dennis Leventhal and Adrian S. Lewis. "Randomized Methods for Linear Constraints: Convergence Rates and Conditioning". In: *Mathematics of Operations Research* 35.3 (2010).

[32] Qihang Lin, Zhaosong Lu, and Lin Xiao. "An Accelerated Proximal Coordinate Gradient Method". In: *NIPS*. 2014.

[33] Ji Liu and Stephen J. Wright. "An Accelerated Randomized Kaczmarz Algorithm". In: *Mathematics of Computation* 85.297 (2016).

[34] Yuanyuan Liu, Fanhua Shang, James Cheng, Hong Cheng, and Licheng Jiao. "Accelerated First-order Methods for Geodesically Convex Optimization on Riemannian Manifolds". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. 2017, pp. 4868–4877.

[35] Zhaosong Lu and Lin Xiao. "On the Complexity Analysis of Randomized Block-Coordinate Descent Methods". In: *Mathematical Programming* 152.1–2 (2015).

[36] A. M. Lyapunov and A. T. Fuller. *General Problem of the Stability Of Motion*. 1992.

[37] Panayotis Mertikopoulos and Mathias Staudigl. *Stochastic mirror descent dynamics and their convergence in monotone variational inequalities*. Arxiv preprint arXiv. 2017.

[38] Deanna Needell and Joel A. Tropp. "Paved with Good Intentions: Analysis of a Randomized Block Kaczmarz Method". In: *Linear Algebra and its Applications* 441 (2014).

[39] Arkadi Nemirovskii and David Yudin. *Problem Complexity and Method Efficiency in Optimization*. John Wiley & Sons, 1983.

[40] Yurii Nesterov. "Accelerating the cubic regularization of Newton's method on convex problems". In: *Mathematical Programming* 112.1 (2008), pp. 159–181.

[41] Yurii Nesterov. *Complexity bounds for primal-dual methods minimizing the model of objective function*. Tech. rep. Université Catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2015.

[42] Yurii Nesterov. "Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems". In: *SIAM J. Optim.* 22.2 (2012).

[43] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization. Boston: Kluwer, 2004.

[44] Yurii Nesterov. "Primal-dual subgradient methods for convex problems". In: *Mathematical Programming* 120.1 (2009), pp. 221–259.

[45] Yurii Nesterov. "Smooth Minimization of Non-smooth Functions". In: *Mathematical Programming* 103.1 (2005), pp. 127–152.

[46] Yurii Nesterov. "Universal gradient methods for convex optimization problems". In: *Mathematical Programming* (2014), pp. 1–24.

[47] Yurii Nesterov and Vladimir Shikhman. "Quasi-monotone Subgradient Methods for Nonsmooth Convex Minimization". In: *Journal of Optimization Theory and Applications* 165.3 (2015), pp. 917–940.

[48] Yurii Nesterov and Sebastian Stich. *Efficiency of Accelerated Coordinate Descent Method on Structured Optimization Problems*. Tech. rep. Université catholique de Louvain, CORE Discussion Papers, 2016.

[49] Julie Nutini, Mark Schmidt, Issam H. Laradji, Michael Friedlander, and Hoyt Koepke. "Coordinate Descent Converges Faster with the Gauss-Southwell Rule Than Random Selection". In: *ICML*. 2015.

[50] Julie Nutini et al. "Convergence Rates for Greedy Kaczmarz Algorithms, and Faster Randomized Kaczmarz Rules Using the Orthogonality Graph". In: *UAI*. 2016.

[51] Bernt Oksendal. *Stochastic Differential Equations (3rd Ed.): An Introduction with Applications*. New York, NY, USA: Springer-Verlag New York, Inc., 1992.

[52] *On Symplectic Optimization*. Arxiv preprint arXiv1802.03653. 2018.

[53] Neal Parikh and Stephen P. Boyd. "Proximal Algorithms". In: *Foundations and Trends in Optimization* 1.3 (2014), pp. 127–239.

[54] Boris T. Polyak. "Some methods of speeding up the convergence of iteration methods". In: *USSR Computational Mathematics and Mathematical Physics* 4.5 (1964), pp. 1–17.

[55] Zheng Qu and Peter Richtárik. "Coordinate Descent with Arbitrary Sampling I: Algorithms and Complexity". In: *arXiv* 1412.8060 (2014).

[56] Zheng Qu and Peter Richtárik. "Coordinate Descent with Arbitrary Sampling II: Expected Separable Overapproximation". In: *arXiv* 1412.8063 (2014).

[57] Zheng Qu, Peter Richtárik, Martin Takáč, and Olivier Fercoq. "SDNA: Stochastic Dual Newton Ascent for Empirical Risk Minimization". In: *ICML*. 2016.

[58] Zheng Qu, Peter Richtárik, and Tong Zhang. "Randomized Dual Coordinate Ascent with Arbitrary Sampling". In: *NIPS*. 2015.

[59] Maxim Raginsky and Jake V. Bouvrie. "Continuous-time stochastic Mirror Descent on a network: Variance reduction, consensus, convergence". In: *Proceedings of the 51th IEEE Conference on Decision and Control, CDC 2012, December 10-13, 2012, Maui, HI, USA*. 2012, pp. 6793–6800.

[60] Ali Rahimi and Benjamin Recht. "Random Features for Large-Scale Kernel Machines". In: *NIPS*. 2007.

[61] Garvesh Raskutti and Sayan Mukherjee. "The Information Geometry of Mirror Descent". In: *IEEE Transactions on Information Theory* 61.3 (2015), pp. 1451–1457.

[62] Benjamin Recht and Christopher Ré. "Parallel Stochastic Gradient Algorithms for Large-Scale Matrix Completion". In: *Mathematical Programming Computation* 5.2 (2013), pp. 201–226.

[63] Peter Richtárik and Martin Takáč. "Iteration Complexity of Randomized Block-Coordinate Descent Methods for Minimizing a Composite Function". In: *Mathematical Programming* 114 (1 2014).

[64] R. Tyrrell Rockafellar. *Convex analysis*. Princeton Mathematical Series. Princeton University Press, 1970.

[65] Alessandro Rudi, Luigi Carratino, and Lorenzo Rosasco. "FALKON: An Optimal Large Scale Kernel Method". In: *arXiv* 1705.10958 (2017).

[66] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors". In: *Nature* 323.6088 (9, 1986), pp. 533–536.

[67] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels*. MIT Press, 2001.

[68] Evan R. Sparks, Shivaram Venkataraman, Tomer Kaftan, Michael Franklin, and Benjamin Recht. "KeystoneML: Optimizing Pipelines for Large-Scale Advanced Analytics". In: *ICDE*. 2017.

[69] Thomas Strohmer and Roman Vershynin. "A Randomized Kaczmarz Algorithm with Exponential Convergence". In: *Journal of Fourier Analysis and Applications* 15.1 (2009).

[70] Weijie Su, Stephen Boyd, and Emmanuel J. Candès. "A Differential Equation for Modeling Nesterov's Accelerated Gradient Method: Theory and Insights". In: *Advances in Neural Information Processing Systems (NIPS) 27*. 2014.

[71] Paul Tseng and Sangwoon Yun. "A Coordinate Gradient Descent Method for Nonsmooth Separable Minimization". In: *Mathematical Programming* 117.1 (2009).

[72] Stephen Tu, Rebecca Roelofs, Shivaram Venkataraman, and Benjamin Recht. "Large Scale Kernel Learning using Block Coordinate Descent". In: *arXiv* 1602.05310 (2016).

[73] Stephen Tu et al. "Breaking Locality Accelerates Block Gauss-Seidel". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. 2017, pp. 3482–3491.

[74] Andre Wibisono. *Sampling as optimization in the space of measures: The Langevin dynamics as a composite optimization problem*. Arxiv preprint arXiv1802.08089.

[75] Andre Wibisono and Ashia Wilson. *On Accelerated Methods in Optimization*. Arxiv preprint arXiv1509.03616.

[76] Andre Wibisono, Ashia C. Wilson, and Michael I. Jordan. "A variational perspective on accelerated methods in optimization". In: *Proceedings of the National Academy of Sciences* 113.47 (2016), E7351–E7358.

[77] Ashia C. Wilson, Benjamin Recht, and Michael I. Jordan. "A Lyapunov Analysis of Momentum Methods in Optimization". In: *arXiv* 1611.02635 (2016).

[78] Fuzhen Zhang. *The Schur Complement and its Applications*. Vol. 4. Numerical Methods and Algorithms. Springer, 2005.

[79] Hongyi Zhang and Suvrit Sra. "First-order Methods for Geodescially Convex Optimization". In: *Conference on Learning Theory (COLT)* (2016).