CS395T: Continuous Algorithms, Part V
Acceleration and high-order methods

Kevin Tian

1 Acceleration

In this lecture, we conclude our development on the basic theory of derivative-based optimization for structured functions. We begin by discussing the phenomenon of *acceleration*, a powerful tool introduced by [Nes83] (with an earlier version in [Nem82]). The main sales pitch for acceleration is that methods developed with this technique have resulted in provably optimal algorithms under a variety of oracle access models (e.g., gradient or high-order derivative queries). The general philosophy behind acceleration (maintaining history-dependent updates, or implementing *momentum*) has also been very empirically succesful in training deep neural networks [SMDH13].

Unfortunately, acceleration has garnered a reputation for being rather difficult to understand. There has been significant effort by the community to develop more intuition for how acceleration arises [Har13, BLS15, SBC16] and how to design accelerated algorithms [LRP16, ZO17, CST21].

We present a proof of acceleration which is built up to in several stages, patterned off Parts II and III of the notes. We begin from a continuous perspective (i.e., a *second-order* ODE), implicitly discretize the continuous dynamics to obtain an accelerated proximal point method, and finally fully discretize it to give an explicit gradient-based method. Each stage has relatively short proofs, which we hope grants intuition on how the different parts of the final accelerated method arise.

1.1 Acceleration in continuous time

The basic intuition behind acceleration is that it learns a trajectory over time, by accumulating previously-queried gradients. This can be naturally modeled as a *second-order* differential equation, which maintains an acceleration variable (i.e., the change in velocity) $\ddot{\mathbf{x}}_t := \frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathbf{x}_t$, in addition to a velocity variable $\dot{\mathbf{x}}_t$. Note that in the gradient flow dynamics, we only maintain a velocity variable.

The following accelerated gradient flow dynamics were introduced by [SBC16]:

$$\ddot{\mathbf{x}}_t + \frac{3}{t}\dot{\mathbf{x}}_t + \nabla f(\mathbf{x}_t) = 0. \tag{1}$$

An equivalent way to interpret the dynamics (1) is by decoupling the position and velocity variables:

$$\dot{\mathbf{v}}_t = -\frac{3}{t}\mathbf{v}_t - \nabla f(\mathbf{x}_t),\ \dot{\mathbf{x}}_t = \mathbf{v}_t.$$

We now give a proof that $f(\mathbf{x}_t)$ decays at an accelerated $\frac{1}{t^2}$ rate, following [SBC16].

Proposition 1 (Accelerated gradient flow)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and L-smooth for $L > 0$,*[1] *and let $\mathbf{x}^\star \in \operatorname{argmin}_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})$. For $\mathbf{x}_t$ following the ODE (1) starting from $\mathbf{x}_0 \in \mathbb{R}^d$ and $\dot{\mathbf{x}}_0 = \mathbf{0}_d$,*

$$f(\mathbf{x}_t) - f(\mathbf{x}^\star) \le \frac{2\left\|\mathbf{x}_0 - \mathbf{x}^\star\right\|_2^2}{t^2}.$$

Proof. Our strategy is to prove that $\dot{\Phi}_t \le 0$, where our potential function Φ_t is defined by

$$\Phi_t := t^2\left(f(\mathbf{x}_t) - f(\mathbf{x}^\star)\right) + 2\left\|\mathbf{z}_t - \mathbf{x}^\star\right\|_2^2, \text{ where } \mathbf{z}_t := \mathbf{x}_t + \frac{t}{2}\dot{\mathbf{x}}_t,$$

[1] This is enough to conclude that (1) has a unique solution, by the Picard-Lindelöf theorem.

from which the conclusion follows because $\Phi_t \geq f(\mathbf{x}_t) - f(\mathbf{x}^\star)$ and $\mathbf{z}_0 = \mathbf{x}_0$. We first compute

$$\begin{aligned}\dot{\Phi}_t &= 2t\left(f(\mathbf{x}_t) - f(\mathbf{x}^\star)\right) + t^2\left\langle \nabla f(\mathbf{x}_t), \dot{\mathbf{x}}_t\right\rangle + 2\left\langle \mathbf{z}_t - \mathbf{x}^\star, 3\dot{\mathbf{x}}_t + t\ddot{\mathbf{x}}_t\right\rangle \\ &= 2t\left(f(\mathbf{x}_t) - f(\mathbf{x}^\star)\right) + t^2\left\langle \nabla f(\mathbf{x}_t), \dot{\mathbf{x}}_t\right\rangle - 2t\left\langle \nabla f(\mathbf{x}_t), \mathbf{z}_t - \mathbf{x}^\star\right\rangle \\ &= 2t\left(f(\mathbf{x}_t) - f(\mathbf{x}^\star) - \left\langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star\right\rangle\right),\end{aligned}$$

where we substituted (1) in the second line. Our claim $\dot{\Phi}_t \leq 0$ now follows by convexity. □

While Proposition 1 is a remarkably short proof-of-concept that acceleration is achievable, it is somewhat magical. One takeaway is that it is useful to design a "fast-forwarded" trajectory $\mathbf{z}_t$, which induces cancellations in potential functions due to the second-order nature of (1). A mystery that arises is the specific choice of the constant 3 in (1). There is discussion in Section 4, [SBC16] on a phase transition that arises around this constant based on the damping behavior of the ODE.

We present an alternative convergence guarantee for a variant of accelerated gradient flow in the well-conditioned regime (cf. Section 4, Part II), i.e., assuming that the function of interest f is strongly convex. By the reduction in Lemma 11, Part II between the smooth and well-conditioned regimes, the rates achieved by Proposition 1 and Proposition 2 are analogous.

Proposition 2 (Accelerated gradient flow, well-conditioned regime)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be L-smooth and μ-strongly convex, let $\kappa := \frac{L}{\mu}$, and let $\mathbf{x}^\star \in \operatorname{argmin}_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})$. For $\mathbf{x}_t$ following the ODE*

$$\ddot{\mathbf{x}}_t + \frac{2}{\sqrt{\kappa}}\dot{\mathbf{x}}_t + \frac{1}{L}\nabla f(\mathbf{x}_t) = 0, \tag{2}$$

from $\mathbf{x}_0 \in \mathbb{R}^d$ and $\dot{\mathbf{x}}_0 = \mathbf{0}_d$,

$$f(\mathbf{x}_t) - f(\mathbf{x}^\star) \leq 2\exp\left(-\frac{t}{\sqrt{\kappa}}\right)\left(f(\mathbf{x}_0) - f(\mathbf{x}^\star)\right).$$

Proof. Similarly to the proof of Proposition 1, we rewrite (2) in the following way:

$$\dot{\mathbf{x}}_t + \sqrt{\kappa}\ddot{\mathbf{x}}_t = -\frac{1}{\mu\sqrt{\kappa}}\nabla f(\mathbf{x}_t) - \dot{\mathbf{x}}_t,$$

so that the "fast-forwarded" trajectory $\mathbf{z}_t := \mathbf{x}_t + \sqrt{\kappa}\dot{\mathbf{x}}_t$ satisfies

$$\dot{\mathbf{z}}_t = \dot{\mathbf{x}}_t + \sqrt{\kappa}\ddot{\mathbf{x}}_t = -\frac{1}{\mu\sqrt{\kappa}}\nabla f(\mathbf{x}_t) + \frac{1}{\sqrt{\kappa}}(\mathbf{x}_t - \mathbf{z}_t). \tag{3}$$

The potential we track in this proof is

$$\Phi_t := f(\mathbf{x}_t) - f(\mathbf{x}^\star) + \frac{\mu}{2}\left\|\mathbf{z}_t - \mathbf{x}^\star\right\|_2^2.$$

We claim $\dot{\Phi}_t \leq -\frac{1}{\sqrt{\kappa}}\Phi_t$, from which the conclusion follows from Grönwall's inequality (Fact 1, Part II) and strong convexity, which implies $\Phi_0 = f(\mathbf{x}_0) - f(\mathbf{x}^\star) + \frac{\mu}{2}\left\|\mathbf{x}_0 - \mathbf{x}^\star\right\|_2^2 \leq 2\left(f(\mathbf{x}_0) - f(\mathbf{x}^\star)\right)$.

To prove our claim, we derive

$$\begin{aligned}\dot{\Phi}_t &= \left\langle \nabla f(\mathbf{x}_t), \dot{\mathbf{x}}_t\right\rangle + \mu\left\langle \dot{\mathbf{z}}_t, \mathbf{z}_t - \mathbf{x}^\star\right\rangle \\ &= -\frac{1}{\sqrt{\kappa}}\left(\left\langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{z}_t\right\rangle + \left\langle \nabla f(\mathbf{x}_t) + \mu(\mathbf{z}_t - \mathbf{x}_t), \mathbf{z}_t - \mathbf{x}^\star\right\rangle\right) \\ &= -\frac{1}{\sqrt{\kappa}}\left(\left\langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star\right\rangle + \mu\left\langle \mathbf{z}_t - \mathbf{x}_t, \mathbf{z}_t - \mathbf{x}^\star\right\rangle\right) \\ &\leq -\frac{1}{\sqrt{\kappa}}\left(\left(\left\langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star\right\rangle - \frac{\mu}{2}\left\|\mathbf{x}_t - \mathbf{x}^\star\right\|_2^2\right) + \frac{\mu}{2}\left\|\mathbf{z}_t - \mathbf{x}^\star\right\|_2^2\right) \leq -\frac{1}{\sqrt{\kappa}}\Phi_t.\end{aligned}$$

The second line used our definitions of $\mathbf{z}_t$ and $\dot{\mathbf{z}}_t$ in (3), the first inequality used Eq. (7), Part III and dropped a nonnegative term, and the second inequality used strong convexity of f. □

1.2 Accelerated proximal point

Acceleration is sometimes viewed as arising from a careful "linear coupling" between gradient descent and mirror descent, that benefits from the convergence guarantees of each [ZO17]. Indeed, the key analysis technique from mirror descent (i.e., the three-point equality in Eq. (7), Part III) has already suggestively made an appearance in the proof of Proposition 2.

Before fully discretizing the ODE (1), in this section we present an implicit variant known as the accelerated proximal point algorithm (APPA) [Gü92], in analogy to the proximal point method of Section 2, Part III. We choose this presentation format for a few reasons, e.g., it leads to a slightly simpler proof, and is more consistent with our later development in Section 2 for accelerated high-order methods. Moreover, by comparing this section with our final method in Section 1.3, the role of gradient descent under the linear coupling perspective becomes more clear.

The APPA is initialized with $(A_0, \mathbf{x}_0) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^d$, and is driven by step sizes $\{\lambda_k\}_{k \geq 1} \subset \mathbb{R}_{\geq 0}$. It evolves three sequences $\{\mathbf{x}_k\}_{k \geq 0}$, $\{\mathbf{y}_k\}_{k \geq 0}$, $\{\mathbf{z}_k\}_{k \geq 0}$, initialized at $\mathbf{z}_0 = \mathbf{y}_0 \leftarrow \mathbf{x}_0$, as follows:

$$\begin{aligned} \mathbf{y}_k &\leftarrow \frac{A_k}{A_{k+1}} \mathbf{x}_k + \frac{a_{k+1}}{A_{k+1}} \mathbf{z}_k, \\ \mathbf{z}_{k+1} &\leftarrow \mathbf{z}_k - a_{k+1} \nabla f(\mathbf{x}_{k+1}), \\ \mathbf{x}_{k+1} &\leftarrow \mathbf{y}_k - \lambda_{k+1} \nabla f(\mathbf{x}_{k+1}), \end{aligned} \tag{4}$$

for all $k \geq 0$, where

$$A_{k+1} \leftarrow A_k + a_{k+1}, \; a_{k+1} \leftarrow \frac{\lambda_{k+1} + \sqrt{\lambda_{k+1}^2 + 4\lambda_{k+1} A_k}}{2}. \tag{5}$$

At this point, it is helpful to provide some intuition on the form of (4), (5), so they do not seem to appear out of nowhere. First, observe that if we fix A_0 and the step size sequence $\{\lambda_k\}_{k \geq 1}$, the sequences $\{A_k, a_k\}_{k \geq 1}$ are uniquely induced via (5). Moreover, by the quadratic formula, we have from (5) that the following recurrence relation holds for all $k \geq 0$:

$$a_{k+1}^2 - \lambda_{k+1} a_{k+1} - \lambda_{k+1} A_k = a_{k+1}^2 - \lambda_{k+1} A_{k+1} = 0 \implies a_{k+1}^2 = \lambda_{k+1} A_{k+1}. \tag{6}$$

The form of the sequences in (4) has a simple intuition. The sequence of $\{\mathbf{x}_k\}_{k \geq 1}$ evolve via the proximal point method (Theorem 1, Part III with $\mathcal{X} = \mathbb{R}^d$ and $\varphi = \frac{1}{2} \|\cdot\|_2^2$) using step sizes $\{\lambda_k\}_{k \geq 1}$. Conversely, the sequence of $\{\mathbf{z}_k\}_{k \geq 1}$ evolve via mirror descent (Theorem 2, Part III), using step sizes $\{a_k\}_{k \geq 1}$ and linear functions induced via the $\{\nabla f(\mathbf{x}_k)\}_{k \geq 1}$.

Finally, the sequence of $\{\mathbf{y}_k\}_{k \geq 1}$ are formed by convex combinations of the other two sequences. For this reason, APPA can be viewed as a "linear coupling" of the proximal point method and mirror descent, foreshadowing a similar linear coupling interpretation of our final accelerated method.

The main invariant satisfied by the iterations (4), (5) is given by the following technical lemma.

Lemma 1. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable and convex. Following the notation (4), (5), define*

$$\Phi_k := A_k \epsilon_k + r_k, \;\; \textit{where } \epsilon_k := f(\mathbf{x}_k) - f(\mathbf{x}^\star), \;\; \textit{and } r_k := \frac{1}{2} \|\mathbf{z}_k - \mathbf{x}^\star\|_2^2, \tag{7}$$

for all $k \geq 0$, where $\mathbf{x}^\star := \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x})$. Then for all $k \geq 0$, $\Phi_{k+1} \leq \Phi_k$.

Proof. By the definition of $\mathbf{y}_k$ in (4), we have

$$\begin{aligned} a_{k+1} \langle \nabla f(\mathbf{x}_{k+1}), \mathbf{x}^\star - \mathbf{z}_k \rangle &= a_{k+1} \langle \nabla f(\mathbf{x}_{k+1}), \mathbf{x}^\star - \mathbf{x}_{k+1} \rangle + a_{k+1} \langle \nabla f(\mathbf{x}_{k+1}), \mathbf{x}_{k+1} - \mathbf{z}_k \rangle \\ &= a_{k+1} \langle \nabla f(\mathbf{x}_{k+1}), \mathbf{x}^\star - \mathbf{x}_{k+1} \rangle \\ &\quad + A_k \langle \nabla f(\mathbf{x}_{k+1}), \mathbf{x}_k - \mathbf{x}_{k+1} \rangle + A_{k+1} \langle \nabla f(\mathbf{x}_{k+1}), \mathbf{x}_{k+1} - \mathbf{y}_k \rangle \\ &\leq a_{k+1} \left(f(\mathbf{x}^\star) - f(\mathbf{x}_{k+1}) \right) + A_k \left(f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \right) \\ &\quad + A_{k+1} \langle \nabla f(\mathbf{x}_{k+1}), \mathbf{x}_{k+1} - \mathbf{y}_k \rangle, \end{aligned}$$

where in the only inequality, we applied convexity twice. By substituting the definition of $\mathbf{x}_{k+1}$ from (4), and using the equality (6), we have

$$\begin{aligned} a_{k+1} \langle \nabla f(\mathbf{x}_{k+1}), \mathbf{x}^\star - \mathbf{z}_k \rangle &\le A_k \left(f(\mathbf{x}_k) - f(\mathbf{x}^\star) \right) - A_{k+1} \left(f(\mathbf{x}_{k+1}) - f(\mathbf{x}^\star) \right) \\ &\quad - a_{k+1}^2 \left\| \nabla f(\mathbf{x}_{k+1}) \right\|_2^2 \\ &= A_k \epsilon_k - A_{k+1} \epsilon_{k+1} - a_{k+1}^2 \left\| \nabla f(\mathbf{x}_{k+1}) \right\|_2^2 . \end{aligned} \tag{8}$$

Moreover, by using the standard mirror descent analysis (see, e.g., Eq. (11), Part III) we have

$$\begin{aligned} a_{k+1} \langle \nabla f(\mathbf{x}_{k+1}), \mathbf{z}_k - \mathbf{x}^\star \rangle &\le \frac{1}{2} \left\| \mathbf{z}_k - \mathbf{x}^\star \right\|_2^2 - \frac{1}{2} \left\| \mathbf{z}_{k+1} - \mathbf{x}^\star \right\|_2^2 + \frac{a_{k+1}^2}{2} \left\| \nabla f(\mathbf{x}_{k+1}) \right\|_2^2 \\ &= r_k - r_{k+1} + \frac{a_{k+1}^2}{2} \left\| \nabla f(\mathbf{x}_{k+1}) \right\|_2^2 . \end{aligned} \tag{9}$$

Combining (8) and (9), we have the conclusion:

$$0 \le \Phi_k - \Phi_{k+1} - \frac{a_{k+1}^2}{2} \left\| \nabla f(\mathbf{x}_{k+1}) \right\|_2^2 \le \Phi_k - \Phi_{k+1}. \tag{10}$$

□

At this point, by applying $\Phi_k \le \Phi_0$, it is evident that the function error ϵ_k decreases at a rate proportional to $\frac{1}{A_k}$, so our goal is to choose $\{\lambda_k\}_{k \ge 1}$ so that A_k grows as quickly as possible. In principle, we could simply choose $\lambda_k \to \infty$; however, this will pose issues when we discretize APPA in the following Section 1.3. We provide a simple bound when $\lambda_k \equiv \lambda$ uniformly.

Lemma 2. *Following the notation in (4), (5), suppose that $\lambda_k = \lambda$ for all $k \ge 1$. Then,*

$$A_k \ge \frac{\lambda k^2}{4}.$$

Proof. This follows from the sequence of bounds:

$$\begin{aligned} \sqrt{A_k} = \sqrt{A_k} - \sqrt{A_0} &= \sum_{i \in [k]} \sqrt{A_i} - \sqrt{A_{i-1}} \\ &= \sum_{i \in [k]} \frac{a_i}{\sqrt{A_i} + \sqrt{A_{i-1}}} = \sum_{i \in [k]} \frac{\sqrt{\lambda_i A_i}}{\sqrt{A_i} + \sqrt{A_{i-1}}} \\ &\ge \frac{1}{2} \sum_{i \in [k]} \sqrt{\lambda_i} = \frac{k\sqrt{\lambda}}{2}, \end{aligned}$$

where the second line used (6), and the third line used that the $\{A_k\}_{k \ge 1}$ are nondecreasing. □

By combining Lemma 1 with Lemma 2, we obtain the main result of this section.

Theorem 1 (Accelerated proximal point)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and differentiable, and suppose for $\mathbf{x}_0 \in \mathbb{R}^d$ we have $\left\| \mathbf{x}_0 - \mathbf{x}^\star \right\|_2 \le R$ for $\mathbf{x}^\star \in \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x})$. Further, let $A_0 = 0$ and $\lambda_k = \lambda > 0$ for all $k \ge 1$. Then iterating (4), (5) for $0 \le k < T$,*

$$f(\mathbf{x}_T) - f(\mathbf{x}^\star) \le \frac{2R^2}{\lambda T^2}.$$

Proof. By Lemma 1, we have

$$f(\mathbf{x}_T) - f(\mathbf{x}^\star) = \epsilon_T \le \frac{\Phi_T}{A_T} \le \frac{\Phi_0}{A_T} \le \frac{R^2}{2A_T}.$$

The conclusion follows from Lemma 2. □

Theorem 1 generically improves upon our analysis of smooth gradient descent (Theorem 3, Part II) by a quadratic factor in the dependence on T, as long as $\lambda \ge \frac{1}{L}$. The catch, of course, is that the iteration (4) is implicit: we cannot exactly compute the proximal point sequence $\{\mathbf{x}_k\}_{k \ge 0}$ in general (cf. discussion in Remark 2, Part III). This motivates our development in the following Section 1.3, where for smooth functions f, we show how to match the convergence rate of Theorem 1 (for an appropriate choice of λ) using an explicit first-order method.

1.3 Accelerated gradient descent

In this section we finally show how to fully discretize the accelerated gradient flow (1) to obtain Nesterov's accelerated gradient descent (AGD) algorithm. In fact, we have already seen the main pieces that we need to put together. Roughly speaking, the key observation is that there is slack in the proof of Lemma 2, in the form of a squared gradient norm (see (10)). We use this slack to compensate for the discretization error that occurs when we take a gradient descent step in place of the proximal point iteration used to define (4), using Corollary 2, Part II.

We again initialize AGD with $(A_0, \mathbf{x}_0) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^d$. In this section, we uniformly set $\lambda_k = \frac{1}{L}$ for all $k \geq 1$, where L is the smoothness of f that we wish to minimize. We again initialize our sequences $\{\mathbf{x}_k\}_{k\geq 0}$, $\{\mathbf{y}_k\}_{k\geq 0}$, $\{\mathbf{z}_k\}_{k\geq 0}$ from $\mathbf{z}_0 = \mathbf{y}_0 \leftarrow \mathbf{x}_0$, and evolve them very similarly to (4):

$$\begin{aligned} \mathbf{y}_k &\leftarrow \frac{A_k}{A_{k+1}}\mathbf{x}_k + \frac{a_{k+1}}{A_{k+1}}\mathbf{z}_k, \\ \mathbf{z}_{k+1} &\leftarrow \mathbf{z}_k - a_{k+1}\nabla f(\mathbf{y}_k), \\ \mathbf{x}_{k+1} &\leftarrow \mathbf{y}_k - \frac{1}{L}\nabla f(\mathbf{y}_k), \end{aligned} \tag{11}$$

for all $k \geq 0$, where $\{A_k, a_k\}_{k\geq 1}$ again follow the recursion (5). That is, compared to (4), (11) is exactly the same except it uses the (explicit) gradients computed at the previous point $\mathbf{y}_k$, rather than the (implicit) gradient computed via the proximal point iterate $\mathbf{x}_{k+1}$.

As we can see, (11) performs a linear coupling of mirror and gradient descent, just as (4) linearly coupled mirror descent and the proximal point method. The intuition provided in [ZO17] is that mirror descent works well when gradients are small (as reflected in the Lipschitz parameter arising in Theorem 2, Part III), and gradient descent works well when gradients are large (as seen in the progress bound in Corollary 2, Part II). Thus, (11) balances the benefits of each method.

We produce the analog of Lemma 1 in the setting of AGD.

Lemma 3. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be L-smooth and convex. Following the notation in (5), (7), (11), and letting $\lambda_k = \frac{1}{L}$ for all $k \geq 1$, we have for all $k \geq 0$ that $\Phi_{k+1} \leq \Phi_k$.*

Proof. Replicating the proof of Lemma 1, we first derive

$$\begin{aligned} a_{k+1}\left\langle \nabla f(\mathbf{y}_k), \mathbf{x}^\star - \mathbf{z}_k \right\rangle &= a_{k+1}\left\langle \nabla f(\mathbf{y}_k), \mathbf{x}^\star - \mathbf{y}_k \right\rangle + a_{k+1}\left\langle \nabla f(\mathbf{y}_k), \mathbf{y}_k - \mathbf{z}_k \right\rangle \\ &= a_{k+1}\left\langle \nabla f(\mathbf{y}_k), \mathbf{x}^\star - \mathbf{y}_k \right\rangle + A_k\left\langle \nabla f(\mathbf{y}_k), \mathbf{x}_k - \mathbf{y}_k \right\rangle \\ &\leq a_{k+1}\left(f(\mathbf{x}^\star) - f(\mathbf{y}_k)\right) + A_k\left(f(\mathbf{x}_k) - f(\mathbf{y}_k)\right), \end{aligned}$$

where again we used the definition of $\mathbf{y}_k$ and convexity twice. Next, analogously to (8), we have

$$\begin{aligned} a_{k+1}\left\langle \nabla f(\mathbf{y}_k), \mathbf{z}_k - \mathbf{x}^\star \right\rangle &\leq r_k - r_{k+1} + \frac{a_{k+1}^2}{2}\left\|\nabla f(\mathbf{y}_k)\right\|_2^2 \\ &\leq r_k - r_{k+1} + a_{k+1}^2 L\left(f(\mathbf{y}_k) - f(\mathbf{x}_{k+1})\right) \\ &= r_k - r_{k+1} + A_{k+1}\left(f(\mathbf{y}_k) - f(\mathbf{x}_{k+1})\right), \end{aligned}$$

where in the second line, we used the progress of smooth gradient descent (Corollary 2, Part II), and in the last line, we used (6) and $\lambda_{k+1} = \frac{1}{L}$. □

We have arrived at our main AGD convergence result, whose proof is identical to Theorem 1.

Theorem 2 (Accelerated gradient descent)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be L-smooth and convex, and suppose for $\mathbf{x}_0 \in \mathbb{R}^d$ we have $\left\|\mathbf{x}_0 - \mathbf{x}^\star\right\|_2 \leq R$ for $\mathbf{x}^\star \in \operatorname{argmin}_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})$. Further, let $A_0 = 0$ and $\lambda_k = \frac{1}{L} > 0$ for all $k \geq 1$. Then iterating (5), (11) for $0 \leq k < T$,*

$$f(\mathbf{x}_T) - f(\mathbf{x}^\star) \leq \frac{2LR^2}{T^2}.$$

Theorem 2 generically improves Theorem 3, Part II, and (via the reduction in Lemma 11, Part II) achieves the tight rate for smooth and well-conditioned convex optimization via gradient queries.

We note that there is a rewriting of the updates (11) commonly seen in derivations of AGD that has a rather intuitive interpretation. In particular, observe that by using (6),

$$\begin{aligned}\mathbf{z}_{k+1} &= \mathbf{z}_k + a_{k+1}L\left(\mathbf{x}_{k+1} - \mathbf{y}_k\right) \\ &= \left(\mathbf{y}_k + \frac{A_k}{a_{k+1}}\left(\mathbf{y}_k - \mathbf{x}_k\right)\right) + \frac{A_{k+1}}{a_{k+1}}\left(\mathbf{x}_{k+1} - \mathbf{y}_k\right) = \mathbf{x}_{k+1} + \frac{A_k}{a_{k+1}}\left(\mathbf{x}_{k+1} - \mathbf{x}_k\right),\end{aligned}$$

so that the update (11) is equivalent to iterating

$$\mathbf{x}_{k+1} \leftarrow \mathbf{y}_k - \frac{1}{L}\nabla f(\mathbf{y}_k),\ \mathbf{y}_{k+1} \leftarrow \mathbf{x}_{k+1} + \frac{A_k}{A_{k+1}}\left(\mathbf{x}_{k+1} - \mathbf{x}_k\right). \tag{12}$$

The iteration (12) explains why acceleration is also often referred to as "momentum," as it can be more concisely described by two update sequences, one of which is advanced via gradient descent, and the other of which is advanced via a history-dependent difference sequence. We also see how this momentum update reflects a discretization of our accelerated gradient flow ODE (1), i.e.,

$$\mathbf{y}_{k+1} \leftarrow \mathbf{y}_k + \frac{A_k}{A_{k+1}}(\mathbf{x}_{k+1} - \mathbf{x}_k) - \frac{1}{L}\nabla f(\mathbf{y}_k),$$

where the two update terms correspond to a momentum term (i.e., accumulating part of the current velocity), and a gradient term, just as in (1). We remark that we are not aware of a "one-sequence" iteration that achieves the tight accelerated rate, so advancing two different sequences of iterates in the discretization of (1) may be inherent. However, there has been interesting recent work on beating the $\frac{1}{T}$ rate of standard gradient descent for smooth, convex functions using alternative discretization techniques such as choosing a careful sequence of step sizes [Gri24, AP23].

Finally, we briefly describe when acceleration is possible in non-Euclidean settings. The conventional wisdom in this regard is that there are two criteria that must be met: to achieve a $\frac{1}{T^2}$ rate for minimizing convex $f : \mathcal{X} \to \mathbb{R}$, we should require that f is L-smooth with respect to some norm $\|\cdot\|$, and that there exists a "small" regularizer $\varphi : \mathcal{X} \to \mathbb{R}$ that is 1-strongly convex with respect to the same norm. For an upper bound achieving this rate, see Theorem 4.1 of [ZO17]. More formally, by small we mean that if φ has an additive range of Θ over $\mathcal{X}$, then the convergence rate scales as $\frac{L\Theta}{T^2}$. Theorem 2 reflects this, where $\Theta = \frac{1}{2}R^2$ for the regularizer $\varphi = \frac{1}{2}\|\cdot\|_2^2$ over $\mathbb{B}(R)$. This poses an issue in norms for which there provably do not exist strongly convex regularizers with additive ranges growing slowly with the dimension d, e.g., the ℓ_∞ norm (cf. Appendix A.1, [ST18]).

One could hope that weaker conditions suffice for acceleration that bypass this strongly convex additive range issue, e.g., that f is relatively smooth in φ (Definition 2, Part III), which in principle could let us design smaller regularizers φ more directly tailored to the geometry of f. Unfortunately, there are lower bounds precluding acceleration in this setting [DTdB22], highlighting that this phenomenon is brittle when extending to non-Euclidean applications.

2 High-order methods

The second topic covered in these notes is *high-order methods*. We consider the setting where our goal is to minimize a p-times differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, and $\nabla^p f$, the p^{th} derivative tensor of f, is Lipschitz. Our notion of Lipschitzness of a tensor-valued function is the following.

Definition 1 (Tensor operator norm)**.** *For $\mathbf{T} \in \mathbb{R}^{d_1 \times d_2 \times \ldots \times d_p}$, we let*

$$\|\mathbf{T}\|_{\text{op}} := \max_{\{\mathbf{v}_i \in \mathbb{R}^{d_i} \mid \|\mathbf{v}_i\|_2 \le 1\}_{i \in [p]}} \left|\mathbf{T}\left[\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p\right]\right|.$$

Definition 2. *Let $\mathbf{T} : \mathbb{R}^d \to \mathbb{R}^{d_1 \times d_2 \times \ldots \times d_p}$. We say that $\mathbf{T}$ is L-Lipschitz if for all $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$,*

$$\|\mathbf{T}(\mathbf{x}) - \mathbf{T}(\mathbf{x}')\|_{\text{op}} \le L\|\mathbf{x} - \mathbf{x}'\|_2.$$

When $\mathbf{T} = \nabla^p f$ for p-times differentiable $f : \mathbb{R}^d \to \mathbb{R}$, we say f has an L-Lipschitz p^{th} derivative.

Observe that when f has an L-Lipschitz p^{th} derivative for $p = 0$, we are simply saying that the function f is Lipschitz. Similarly, when $p = 1$, we are saying that the gradient ∇f is Lipschitz, i.e., that f is smooth, and $p = 2$ means that the Hessian is Lipschitz. To help build intuition for Definitions 1 and 2, we give a characterization of operator norms in the symmetric case.[2]

[2]This applies to derivative tensors $\nabla^p f$ as long as $\nabla^p f$ is continuous, by Schwarz's theorem.

Lemma 4. *Let $\mathbf{T} \in \mathbb{R}^{d^{\otimes p}}$ be a symmetric tensor, i.e., $\mathbf{T}[\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p] = \mathbf{T}[\mathbf{v}_{\pi(1)}, \mathbf{v}_{\pi(2)}, \ldots, \mathbf{v}_{\pi(p)}]$ for any set of vectors $\{\mathbf{v}_i\}_{i\in[p]} \subset \mathbb{R}^d$ and permutation $\pi : [p] \to [p]$. Then,*

$$\|\mathbf{T}\|_{\text{op}} = \max_{\substack{\mathbf{v}\in\mathbb{R}^d \\ \|\mathbf{v}\|_2 \le 1}} \mathbf{T}[\mathbf{v}, \mathbf{v}, \ldots, \mathbf{v}].$$

Proof. The lemma states that the maximum in Definition 1 is achieved by using p copies of the same vector $\mathbf{v}$. It is a well-known fact in linear algebra that the lemma is true when $p = 2$. We sketch the proof for larger p, deferring more details to Appendix 1 of [NN94].

The idea is to show that if there is a vector tuple $\{\mathbf{v}_i\}_{i\in[p]} \in \mathbb{R}^d$ achieving the maximum in Definition 1 so that not all vectors are equal, then we can change the set by appropriately replacing vectors by their average. The basic operation that lets us do this is the observation that $\mathbf{T}[\cdot, \cdot, \mathbf{v}_3, \mathbf{v}_4, \ldots, \mathbf{v}_p]$ is a symmetric matrix for any choices of $\{\mathbf{v}_i\}_{i=3}^p \subset \mathbb{R}^d$. Thus, we can always maximize the total value with a vector tuple satisfying $\mathbf{v}_1 = \mathbf{v}_2$. More generally whenever a maximizing subset does not have this property, we can rotate any disagreeing vectors to the first two spots (by symmetry of $\mathbf{T}$) and "merge" them into two copies of the same vector. The final proof in [NN94] inducts on p, so that the maximizing argument contains at most two types of vectors by passing to the $p-1$ case. It then shows that by repeatedly using the basic operation described above, one can decrease the angle between the two vectors, until they are the same vector. □

In analogy to Lemma 6, Part II, there are a number of equivalent ways to describe a function that has a Lipschitz p^{th} derivative. First, we bound the $(p+1)^{\text{th}}$ derivative when it exists.

Lemma 5. *Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is $(p+1)$-times continuously differentiable. Then f has an L-Lipschitz p^{th} derivative iff for all $\mathbf{x} \in \mathbb{R}^d$, $\|\nabla^{p+1} f(\mathbf{x})\|_{\text{op}} \le L$.*

Proof. To see one direction, we apply Definition 2 with $\mathbf{x}' = \mathbf{x} + t\mathbf{v}$ and vanishing $t \to 0$:

$$\begin{aligned} t^{p+1}\nabla^{p+1} f(\mathbf{x})[\mathbf{v}^{\otimes(p+1)}] + o(t^{p+2}) &= \left\langle \nabla^p f(\mathbf{x}') - \nabla^p f(\mathbf{x}), (\mathbf{x}' - \mathbf{x})^{\otimes p} \right\rangle \\ &\le L \|\mathbf{x}' - \mathbf{x}\|_2^{p+1} = L t^{p+1} \|\mathbf{v}\|_2^{p+1}, \end{aligned}$$

which gives the bound $\nabla^{p+1} f(\mathbf{x})[\mathbf{v}^{\otimes(p+1)}] \le L$, and $\nabla^{p+1} f(\mathbf{x})[\mathbf{v}^{\otimes(p+1)}] \ge -L$ holds similarly. It suffices to bound the operator's value acting upon the same vector $p+1$ times due to Lemma 4.

To see the other direction, denote $\mathbf{x}_\lambda := (1-\lambda)\mathbf{x} + \lambda\mathbf{x}'$ for all $\lambda \in [0, 1]$. Then,

$$\left(\nabla^p f(\mathbf{x}) - \nabla^p f(\mathbf{x}')\right)[\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p] = \int_0^1 \nabla^{p+1} f(\mathbf{x}_\lambda)[\mathbf{x}' - \mathbf{x}, \mathbf{v}_1, \ldots, \mathbf{v}_p] \mathrm{d}\lambda \le L \|\mathbf{x}' - \mathbf{x}\|_2,$$

for any $\{\mathbf{v}_i\}_{i\in[p]}$ with $\|\mathbf{v}_i\|_2 \le 1$ for all $i \in [p]$, if $\|\nabla^{p+1} f(\mathbf{x}_\lambda)\|_{\text{op}} \le L$ for all $\lambda \in [0, 1]$. □

Next, we give a formula for the remainders of higher derivatives. We use the notation that

$$\mathcal{T}^k f_{\bar{\mathbf{x}}}(\mathbf{x}) := \sum_{i=0}^k \frac{1}{i!} \nabla^i f(\bar{\mathbf{x}})[(\mathbf{x} - \bar{\mathbf{x}})^{\otimes i}] \tag{13}$$

is the k^{th}-order Taylor series of f around $\bar{\mathbf{x}}$. For example, $\mathcal{T}^2 f_{\bar{\mathbf{x}}}(\mathbf{x}) = f(\bar{\mathbf{x}}) + \langle \nabla f(\bar{\mathbf{x}}), \mathbf{x} - \bar{\mathbf{x}} \rangle + \frac{1}{2}\nabla^2 f(\bar{\mathbf{x}})[\mathbf{x} - \bar{\mathbf{x}}, \mathbf{x} - \bar{\mathbf{x}}]$ is the commonly-used *Newton approximation* of f.

Lemma 6. *Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is p-times differentiable. Then for all $\bar{\mathbf{x}}, \mathbf{x} \in \mathbb{R}^d$ and all $0 \le i < p$,*

$$\nabla^i f(\mathbf{x}) = \nabla^i \mathcal{T}_{\bar{\mathbf{x}}}^{p-1} f(\mathbf{x}) + \int_0^1 \frac{\lambda^{p-i-1}}{(p-i-1)!} \nabla^p f(\mathbf{x}_\lambda)[(\mathbf{x} - \bar{\mathbf{x}})^{\otimes(p-i)}] \mathrm{d}\lambda,$$

where $\mathbf{x}_\lambda := (1-\lambda)\mathbf{x} + \lambda\mathbf{x}'$ for all $\lambda \in [0, 1]$. Moreover, for any $i \in [p]$, as a function of $\mathbf{x}$,

$$\nabla\left(\nabla^i f(\bar{\mathbf{x}})\left[(\mathbf{x} - \bar{\mathbf{x}})^{\otimes i}\right]\right) = i\nabla^i f(\bar{\mathbf{x}})\left[(\mathbf{x} - \bar{\mathbf{x}})^{\otimes(i-1)}, \cdot\right].$$

Proof. This follows from the fundamental theorem of calculus, recursively applied $p - i$ times (see, e.g., the derivation in Lemma 6, Part II for an example of this when $p = 2$ and $i = 0$). □

As a consequence of Lemma 6, having a Lipschitz p^{th} derivative implies the following useful bounds.

Lemma 7. *Let $f : \mathbb{R}^d \to \mathbb{R}$ have an L-Lipschitz p^{th} derivative. Then for all $\mathbf{x}, \bar{\mathbf{x}} \in \mathbb{R}^d$,*

$$\mathcal{T}_{\bar{\mathbf{x}}}^p f(\mathbf{x}) - \frac{L}{(p+1)!} \|\mathbf{x} - \bar{\mathbf{x}}\|_2^{p+1} \le f(\mathbf{x}) \le \mathcal{T}_{\bar{\mathbf{x}}}^p f(\mathbf{x}) + \frac{L}{(p+1)!} \|\mathbf{x} - \bar{\mathbf{x}}\|_2^{p+1}. \tag{14}$$

Moreover, if f is convex, then the upper bound in (14) is also convex.

Proof. For simplicity, we only prove the first claim in the case when f is $(p+1)$-times continuously differentiable; Lemma 11.2.3 of [Sid23] handles the more general case with a slightly more complicated proof. To see the first claim, we use Lemma 6 with $i \leftarrow 0$ and $p \leftarrow p+1$, and bound

$$\int_0^1 \frac{\lambda^{p-i}}{(p-i)!} \nabla^{p+1} f(\mathbf{x}_\lambda)[(\mathbf{x} - \bar{\mathbf{x}})^{\otimes(p+1)}] \mathrm{d}\lambda \le \frac{L}{p!} \|\mathbf{x} - \bar{\mathbf{x}}\|_2^p \int_0^1 \lambda^p \mathrm{d}\lambda,$$

where we applied the operator norm bound from Lemma 5. The conclusion follows because $\int_0^1 \lambda^p \mathrm{d}\lambda = \frac{1}{p+1}$. We remark that the same proof straightforwardly shows that for any $i \in [p]$,

$$\left\|\nabla^i f(\mathbf{x}) - \nabla^i \mathcal{T}_{\bar{\mathbf{x}}}^p f(\mathbf{x})\right\|_{\text{op}} \le \frac{L}{(p+1-i)!} \|\mathbf{x} - \bar{\mathbf{x}}\|_2^{p+1-i}.$$

In particular, applying the above expression with $i = 2$ gives

$$\left\|\nabla^2 f(\mathbf{x}) - \nabla^2 \mathcal{T}_{\bar{\mathbf{x}}}^p f(\mathbf{x})\right\|_{\text{op}} \le \frac{L}{(p-1)!} \|\mathbf{x} - \bar{\mathbf{x}}\|_2^{p-1}. \tag{15}$$

On the other hand, we show that the regularization component of the upper bound in (14) is sufficiently strongly convex. Recalling that the gradient of $\|\mathbf{v}\|_2$ is $\frac{\mathbf{v}}{\|\mathbf{v}\|_2}$, we compute

$$\begin{aligned}
\nabla^2 \left(\frac{L}{(p+1)!} \|\mathbf{x} - \bar{\mathbf{x}}\|_2^{p+1} \right) &= \nabla \left(\frac{L}{p!} \|\mathbf{x} - \bar{\mathbf{x}}\|_2^p (\mathbf{x} - \bar{\mathbf{x}}) \right) \\
&= \frac{L}{(p-1)!} \|\mathbf{x} - \bar{\mathbf{x}}\|_2^{p-1} \mathbf{I}_d + C(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^\top \\
&\succeq \frac{L}{(p-1)!} \|\mathbf{x} - \bar{\mathbf{x}}\|_2^{p-1} \mathbf{I}_d,
\end{aligned} \tag{16}$$

for some $C > 0$. Convexity of the upper bound in (14) follows from (15) and (16). □

In the remainder of this section, we show how to use the upper bound in Lemma 7 to design *Taylor descent* iterative methods, generalizing the gradient descent algorithms of Part II.

Remark 1. *In principle, Lemma 7 combined with, e.g., Theorem 1, Part I show that we can minimize the right-hand side of (14) to high accuracy in polynomial time, as long as f is convex. However, if minimizing this expression is more expensive than minimizing the function f itself, then this defeats the point of using Lemma 7 in a descent-like procedure. For the specific case of $p = 2$, by using Lagrange multipliers to characterize the minimizer as being of the form*

$$\bar{\mathbf{x}} - \left(\nabla^2 f(\bar{\mathbf{x}}) + \lambda \mathbf{I}_d\right)^{-1} \nabla f(\bar{\mathbf{x}}),$$

for some $\lambda > 0$, [NP06] shows that binary searching efficiently computes the minimizer to high accuracy, even for nonconvex f. This was extended by [CD20] to work using only queries to ∇f. To our knowledge, optimizing the upper bound in (14) for $p > 2$ is not as well-understood.

2.1 Stationary points

As a warmup, we first give the high-order analog of Lemma 7, Part II, which uses gradient queries to efficiently find an ϵ-approximate stationary point of f (i.e., where $\|\nabla f(\mathbf{x})\|_2 \le \epsilon$). Our algorithm will simply repeatedly minimize (a slight modification of) the upper bound in (14).

Lemma 8. *Let $f : \mathbb{R}^d \to \mathbb{R}$ have an L-Lipschitz p^{th} derivative. For any $\bar{\mathbf{x}} \in \mathbb{R}^d$, letting*

$$\mathbf{x} \leftarrow \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^d} \mathcal{T}^p_{\bar{\mathbf{x}}} f(\mathbf{x}) + \frac{2L}{(p+1)!} \|\mathbf{x} - \bar{\mathbf{x}}\|_2^{p+1}, \tag{17}$$

we have

$$f(\mathbf{x}) \leq f(\bar{\mathbf{x}}) - \frac{L}{(p+1)!} \left(\frac{\|\nabla f(\bar{\mathbf{x}})\|_2 \, p!}{3L} \right)^{\frac{p+1}{p}}.$$

Proof. First, because $\mathcal{T}^p_{\bar{\mathbf{x}}} f(\bar{\mathbf{x}}) = f(\bar{\mathbf{x}})$, we have by optimality of $\mathbf{x}$ that

$$\begin{aligned} f(\mathbf{x}) + \frac{L}{(p+1)!} \|\mathbf{x} - \bar{\mathbf{x}}\|_2^{p+1} &\leq \mathcal{T}^p_{\bar{\mathbf{x}}} f(\mathbf{x}) + \frac{2L}{(p+1)!} \|\mathbf{x} - \bar{\mathbf{x}}\|_2^{p+1} \\ &\leq \mathcal{T}^p_{\bar{\mathbf{x}}} f(\bar{\mathbf{x}}) + \frac{2L}{(p+1)!} \|\bar{\mathbf{x}} - \bar{\mathbf{x}}\|_2^{p+1} = f(\bar{\mathbf{x}}). \end{aligned} \tag{18}$$

where the first inequality used the upper bound in Lemma 7. Next, let

$$U_{\bar{\mathbf{x}}} := \mathcal{T}^p_{\bar{\mathbf{x}}} f + \frac{2L}{(p+1)!} \|\cdot - \bar{\mathbf{x}}\|_2^{p+1}$$

be the function minimized by $\mathbf{x}$, so $\nabla U_{\bar{\mathbf{x}}}(\mathbf{x}) = \mathbf{0}_d$. We have

$$\begin{aligned} \|\nabla f(\mathbf{x})\|_2 &\leq \|\nabla f(\mathbf{x}) - \nabla U_{\bar{\mathbf{x}}}(\mathbf{x})\|_2 + \|\nabla U_{\bar{\mathbf{x}}}(\mathbf{x})\|_2 \\ &\leq \|\nabla f(\mathbf{x}) - \nabla \mathcal{T}^p_{\bar{\mathbf{x}}} f(\mathbf{x})\|_2 + \frac{2L}{p!} \|\mathbf{x} - \bar{\mathbf{x}}\|_2^p, \end{aligned}$$

where in both lines, we applied the triangle inequality. By using Lemma 6 with $i \leftarrow 1$ and $p \leftarrow p+1$,

$$\begin{aligned} \|\nabla f(\mathbf{x}) - \nabla \mathcal{T}^p_{\bar{\mathbf{x}}} f(\mathbf{x})\|_2 &\leq \int_0^1 \frac{\lambda^{p-1}}{(p-1)!} \left\| \nabla^{p+1} f(\mathbf{x}_\lambda)[(\mathbf{x} - \bar{\mathbf{x}})^{\otimes p}, \cdot] \right\|_2 \mathrm{d}\lambda \\ &\leq \frac{L}{(p-1)!} \|\mathbf{x} - \bar{\mathbf{x}}\|_2^p \int_0^1 \lambda^{p-1} \mathrm{d}\lambda = \frac{L}{p!} \|\mathbf{x} - \bar{\mathbf{x}}\|_2^p. \end{aligned}$$

Combining thus shows

$$\|\nabla f(\mathbf{x})\|_2 \leq \frac{3L}{p!} \|\mathbf{x} - \bar{\mathbf{x}}\|_2^p,$$

and the conclusion follows by plugging the above display back into (18). □

At this point, repeatedly iterating Lemma 8 yields an approximate stationary point algorithm that converges at the rate of $T^{-\frac{p}{p+1}}$, which improves upon the $T^{-\frac{1}{2}}$ rate achieved for smooth functions in Lemma 7, Part II, for $p > 1$. In the limit of large p, this gives a T^{-1} rate.

Corollary 1. *In the setting of Lemma 8, let $\epsilon > 0$, and suppose for $\mathbf{x}_0 \in \mathbb{R}^d$ we have $f(\mathbf{x}) - \min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}) \leq \Delta$. Then iterating $(\mathbf{x}_{t+1}, \mathbf{x}_t) \leftarrow (\mathbf{x}, \bar{\mathbf{x}})$ as given by (17) for $0 \leq t < T$ where*

$$T \geq \frac{\Delta}{\frac{L}{(p+1)!} \cdot \left(\frac{p!}{3L} \right)^{\frac{p+1}{p}}} \cdot \frac{1}{\epsilon^{\frac{p+1}{p}}},$$

we have

$$\min_{0 \leq t < T} \|\nabla f(\mathbf{x}_t)\|_2 \leq \epsilon.$$

Proof. The proof is identical to Lemma 7, Part II, using the progress bound from Lemma 8. □

In principle, Corollary 1 applies to potentially nonconvex f, just as Lemma 7, Part II did. However, an end-to-end implementation of it requires the ability to actually minimize the objective in (17). For $p = 2$, such methods exist (Remark 1), and for larger p, we can use a lower-order stationary point finder, because Lemma 8 is robust to a small value of $\|\nabla U_{\bar{\mathbf{x}}}(\mathbf{x})\|_2 \neq 0$. We note that these uses are likely much less practical than more direct, structured algorithms when they exist.

2.2 Taylor descent

We now consider the case of minimizing convex f with a Lipschitz p^{th} derivative. Again, we repeatedly minimize the upper bound in (14). This is called *Taylor descent*, and yields the following per-step progress, analyzed similarly to the proximal gradient method in Theorem 7, Part II. This section is based on the exposition in Chapter 11 of [Sid23].

Lemma 9. *Let convex $f : \mathbb{R}^d \to \mathbb{R}$ have an L-Lipschitz p^{th} derivative. For any $\bar{\mathbf{x}} \in \mathbb{R}^d$, letting*

$$\mathbf{x} \leftarrow \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^d} \mathcal{T}^p_{\bar{\mathbf{x}}} f(\mathbf{x}) + \frac{L}{(p+1)!} \|\mathbf{x} - \bar{\mathbf{x}}\|_2^{p+1}, \tag{19}$$

we have for $\mathbf{x}^\star \in \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x})$,

$$f(\mathbf{x}) - f(\mathbf{x}^\star) \leq \left(1 - \frac{p}{p+1} \cdot \min\left(1, \left(\frac{p!(f(\bar{\mathbf{x}}) - f(\mathbf{x}^\star))}{2L\|\bar{\mathbf{x}} - \mathbf{x}^\star\|_2^{p+1}}\right)^{\frac{1}{p}}\right)\right)(f(\bar{\mathbf{x}}) - f(\mathbf{x}^\star)).$$

Proof. Throughout the proof, let $U_{\bar{\mathbf{x}}} := \mathcal{T}^p_{\bar{\mathbf{x}}} f + \frac{L}{(p+1)!}\|\cdot - \bar{\mathbf{x}}\|_2^{p+1}$, so that $\mathbf{x} = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^d} U_{\bar{\mathbf{x}}}(\mathbf{x})$. Letting $\mathbf{x}_\lambda := \lambda \mathbf{x}^\star + (1-\lambda)\bar{\mathbf{x}}$ for all $\lambda \in [0,1]$, by optimizing over the line between $\bar{\mathbf{x}}$ and $\mathbf{x}^\star$,

$$\begin{aligned} f(\mathbf{x}) - f(\mathbf{x}^\star) &\leq U_{\bar{\mathbf{x}}}(\mathbf{x}) - f(\mathbf{x}^\star) \leq \min_{\lambda \in [0,1]} U_{\bar{\mathbf{x}}}(\mathbf{x}_\lambda) - f(\mathbf{x}^\star) \\ &\leq \min_{\lambda \in [0,1]} f(\mathbf{x}_\lambda) + \frac{2L}{(p+1)!}\|\mathbf{x} - \bar{\mathbf{x}}\|_2^{p+1} - f(\mathbf{x}^\star) \\ &\leq \min_{\lambda \in [0,1]} (1-\lambda)(f(\bar{\mathbf{x}}) - f(\mathbf{x}^\star)) + \frac{2L\lambda^{p+1}}{(p+1)!}\|\bar{\mathbf{x}} - \mathbf{x}^\star\|_2^{p+1}, \end{aligned} \tag{20}$$

where the first line uses the upper bound in Lemma 7, the second uses the lower bound in Lemma 7, and the last applies convexity of f. Next, note that for $\alpha, \beta \geq 0$,

$$\min_{\lambda \in [0,1]} -\alpha t + \frac{\beta \lambda^{p+1}}{p+1} = \begin{cases} -\frac{p\alpha}{p+1} \cdot \left(\frac{\alpha}{\beta}\right)^{\frac{1}{p}} & \left(\frac{\alpha}{\beta}\right)^{\frac{1}{p}} \in [0,1] \\ -\frac{p\alpha}{p+1} & \text{else} \end{cases},$$

which can be verified by direct computation of the constrained minimizer. The claim follows from applying the above display to the relevant choices of α and β in (20). □

By iterating on Lemma 9, combined with the use of a differential inequality to aggregate progress bounds, we derive a convergence rate for Taylor descent.

Theorem 3 (Taylor descent)**.** *Let convex $f : \mathbb{R}^d \to \mathbb{R}$ have an L-Lipschitz p^{th} derivative, and suppose for $\mathbf{x}_0 \in \mathbb{R}^d$, $\max_{\mathbf{x} \in \mathbb{R}^d, f(\mathbf{x}) - f(\mathbf{x}_0)} \|\mathbf{x} - \mathbf{x}^\star\|_2 \leq R$ for $\mathbf{x}^\star \in \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x})$. Then iterating*

$$\mathbf{x}_{t+1} \leftarrow \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^d} \mathcal{T}^p_{\bar{\mathbf{x}}} f(\mathbf{x}) + \frac{L}{(p+1)!}\|\mathbf{x} - \bar{\mathbf{x}}\|_2^{p+1}$$

for $0 \leq t < T$, we have

$$f(\mathbf{x}_T) - f(\mathbf{x}^\star) \leq \frac{(p+1)^p}{p!} \cdot \frac{2LR^{p+1}}{(T-1)^p}.$$

Proof. Throughout the proof, let $\Phi_t := f(\mathbf{x}_t) - f(\mathbf{x}^\star)$ for all $0 \leq t \leq T$, and note that by Lemma 9, $\Phi_t \leq \Phi_{t-1}$ for all $t \in [T]$, i.e., Taylor descent is in fact a descent method. Moreover, by Lemma 7,

$$\begin{aligned} f(\mathbf{x}_1) &\leq \mathcal{T}^p_{\mathbf{x}_0} f(\mathbf{x}_1) + \frac{L}{(p+1)!}\|\mathbf{x}_1 - \mathbf{x}_0\|_2^{p+1} \\ &\leq \mathcal{T}^p_{\mathbf{x}_0} f(\mathbf{x}^\star) + \frac{L}{(p+1)!}\|\mathbf{x}^\star - \mathbf{x}_0\|_2^{p+1} \leq f(\mathbf{x}^\star) + \frac{2LR^{p+1}}{(p+1)!}. \end{aligned}$$

Thus, for all $t \geq 1$, we have by Lemma 9 and the above display that

$$\Phi_{t+1} \leq \Phi_t - \left(\frac{p}{p+1} \cdot \left(\frac{p!}{2L}\right)^{\frac{1}{p}} \cdot \left(\frac{1}{R}\right)^{1+\frac{1}{p}}\right) \cdot \Phi_t^{1+\frac{1}{p}}.$$

Now, rewriting the above as $\Phi_{t+1} \le \Phi_t - \alpha \Phi_t^{1+\frac{1}{p}}$, we have

$$\begin{aligned} -\alpha(T-1) &\ge \sum_{t\in[T-1]} \frac{\Phi_{t+1}-\Phi_t}{\Phi_t^{1+\frac{1}{p}}} \\ &\ge \sum_{t\in[T-1]} \int_{\Phi_t}^{\Phi_{t+1}} \frac{1}{\tau^{1+\frac{1}{p}}} \mathrm{d}\tau = \int_{\Phi_1}^{\Phi_T} \frac{1}{\tau^{1+\frac{1}{p}}} \mathrm{d}\tau \\ &= -p\left(\Phi_T^{-\frac{1}{p}} - \Phi_1^{-\frac{1}{p}}\right) \ge -p\Phi_T^{-\frac{1}{p}}. \end{aligned} \tag{21}$$

By rearranging and plugging in the choice of α, we have the claimed bound. □

Up to constant factors depending on p, Theorem 3 states that repeatedly minimizing a regularized p^{th}-order Taylor expansion yields a T^{-p} rate of convergence under a Lipschitz p^{th} derivative assumption, which dramatically improves upon the rate of standard smooth gradient descent (Theorem 3, Part II). The tradeoff is that we need to assume more regularity on our target function, and that the Taylor descent step is efficiently implementable (e.g., by using Remark 1 for $p = 2$).

2.3 Accelerated Taylor descent

Theorem 3 begs the question: under a Lipschitz p^{th}-derivative assumption on f, and oracle access to its first p derivatives, what is the optimal oracle query complexity? In particular, are there *accelerated high-order methods* (in analogy with Theorem 2), and are they optimal?

In [Nes08], Nesterov gave an algorithm for minimizing a function with a Lipschitz Hessian (i.e., $p = 2$) converging at the rate of T^{-3} in T iterations. This improves upon Theorem 3 by a factor of T^{-1} so it is natural to guess that this is the best possible. Surprisingly, in a breakthrough work [MS13] showed that this is improvable to a convergence rate of $T^{-\frac{7}{2}}$ in T iterations under the same assumptions. This led to a flurry of activity culminating in an algorithm by [GDG+19] that uses T queries to the p^{th} derivative oracle of a function with a Lipschitz p^{th} derivative, and achieves error $\propto T^{-\frac{3p+1}{2}}$, up to a logarithmic factor. The logarithmic factor was removed by [CHJ+22, KG22], and [GN20] gave a matching lower bound. Interestingly, the lower bound construction in [GN20] is essentially an ℓ_p-norm variant of the graph Laplacian quadratic in Theorem 5, Part II.

We briefly describe the near-optimal method of [GDG+19] for completeness, which follows the APPA framework (Section 1.2). Recall from (4) that APPA maintains three sequences $\{\mathbf{x}_k\}_{k\ge0}$, $\{\mathbf{y}_k\}_{k\ge0}$, $\{\mathbf{z}_k\}_{k\ge0}$. The $\{\mathbf{x}_k\}_{k\ge0}$ are updated via the proximal point method, the $\{\mathbf{z}_k\}_{k\ge0}$ via mirror descent, and the $\{\mathbf{y}_k\}_{k\ge0}$ via convex combinations of the other two sequences.

In the p^{th}-order method of [GDG+19], the $\{\mathbf{x}_k\}_{k\ge0}$ sequence is modified to update via Taylor descent. The other two sequences are updated in exactly the same way as before. The key observation is that if the step size parameters (5) are chosen to satisfy

$$\frac{1}{2} \le \lambda_{k+1} \cdot \frac{L\left\|\mathbf{x}_{k+1}-\mathbf{y}_k\right\|^{p-1}}{(p-1)!} \le \frac{p}{p+1}, \tag{22}$$

then one can quickly grow the parameter A_T in T iterations.[3] This is useful because the convergence rate of APPA is related to $\frac{1}{A_T}$ (see the proof of Theorem 1). In our earlier application, we chose a constant value for the $\{\lambda_k\}_{k\ge1}$, and showed this resulted in $A_T \propto T^2$. Under (22), [GDG+19] is able to force A_T to grow much faster via a self-bounded differential inequality, similarly to (21).

Intuitively, the condition (22) is used to show that the update from $\mathbf{y}_k$ to $\mathbf{x}_{k+1}$ is nearly a proximal point step with regularization parameter λ_{k+1}. This is done by comparing the derivatives of the defining objectives using Lemma 6. Under this condition, we essentially have the APPA proof in Theorem 1. However, (22) turns out to give us more fine-grained control of how A_T grows, because of the self-bounded differential inequality it implies. We defer details to [GDG+19].

[3] It is not obvious how to produce λ_{k+1} satisfying (22), since λ_{k+1} is used to define $\mathbf{y}_k$ and $\mathbf{x}_{k+1}$. A binary search is needed to select a step size, leading to the extraneous logarithmic factor in the convergence rate.

Source material

Portions of this lecture are based on reference material in [Nes03, Sid23], as well as the author's own experience working in the field.

References

[AP23] Jason M. Altschuler and Pablo A. Parrilo. Acceleration by stepsize hedging I: multi-step descent and the silver stepsize schedule. *CoRR*, abs/2309.07879, 2023.

[BLS15] Sébastien Bubeck, Yin Tat Lee, and Mohit Singh. A geometric alternative to nesterov's accelerated gradient descent. *CoRR*, abs/1506.08187, 2015.

[CD20] Yair Carmon and John C. Duchi. First-order methods for nonconvex quadratic minimization. *SIAM review*, 62(2):395–436, 2020.

[CHJ+22] Yair Carmon, Danielle Hausler, Arun Jambulapati, Yujia Jin, and Aaron Sidford. Optimal and adaptive monteiro-svaiter acceleration. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022*, 2022.

[CST21] Michael B. Cohen, Aaron Sidford, and Kevin Tian. Relative lipschitzness in extragradient methods and a direct recipe for acceleration. In *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 62:1–62:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[DTdB22] Radu-Alexandru Dragomir, Adrien B. Taylor, Alexandre d'Aspremont, and Jérôme Bolte. Optimal complexity and certification of bregman first-order methods. *Math. Program.*, 194(1):41–83, 2022.

[GDG+19] Alexander V. Gasnikov, Pavel E. Dvurechensky, Eduard Gorbunov, Evgeniya A. Vorontsova, Daniil Selikhanovych, César A. Uribe, Bo Jiang, Haoyue Wang, Shuzhong Zhang, Sébastien Bubeck, Qijia Jiang, Yin Tat Lee, Yuanzhi Li, and Aaron Sidford. Near optimal methods for minimizing convex functions with lipschitz p-th derivatives. In *Conference on Learning Theory, COLT 2019*, volume 99 of *Proceedings of Machine Learning Research*, pages 1392–1393. PMLR, 2019.

[GN20] Geovani Nunes Grapiglia and Yurii Nesterov. Tensor methods for minimizing convex functions with hölder continuous higher-order derivatives. *SIAM Journal on Optimization*, 30(4):2750–2779, 2020.

[Gri24] Benjamin Grimmer. Provably faster gradient descent via long steps. *SIAM Journal on Optimization*, 34(3):2588–2608, 2024.

[Gü92] Osman Güler. New proximal point algorithms for convex minimization. *SIAM Journal on Optimization*, 2(4):649–664, 1992.

[Har13] Moritz Hardt. The zen of gradient descent. http://blog.mrtz.org/2013/09/07/the-zen-of-gradient-descent.html, 2013. Accessed: 2025-02-09.

[KG22] Dmitry Kovalev and Alexander V. Gasnikov. The first optimal acceleration of high-order methods in smooth convex optimization. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022*, 2022.

[LRP16] Laurent Lessard, Benjamin Recht, and Andrew K. Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM J. Optim.*, 26(1):57–95, 2016.

[MS13] Renato D. C. Monteiro and Benar Fux Svaiter. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM Journal on Optimization*, 23(2):1092–1125, 2013.

[Nem82] Arkadi Nemirovski. Orth-method for smooth convex optimization. *Izvestia AN SSSR, Ser. Tekhnicheskaya Kibernetika*, 2, 1982.

[Nes83] Yurii Nesterov. A method for solving a convex programming problem with convergence rate $o(1/k^2)$. *Doklady AN SSSR*, 269:543–547, 1983.

[Nes03] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course, volume I.* 2003.

[Nes08] Yurii Nesterov. Accelerating the cubic regularization of newton's method on convex problems. *Mathematical Programming*, 112:159–181, 2008.

[NN94] Yurii Nesterov and Arkadi Nemirovski. *Interior-Point Polynomial Algorithms in Convex Programming.* Society for Industrial and Applied Mathematics, 1994.

[NP06] Yurii Nesterov and Boris T. Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming, Series A*, 108:177–205, 2006.

[SBC16] Weijie Su, Stephen P. Boyd, and Emmanuel J. Candès. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. *J. Mach. Learn. Res.*, 17:153:1–153:43, 2016.

[Sid23] Aaron Sidford. *Optimization Algorithms.* 2023.

[SMDH13] Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 1139–1147. JMLR.org, 2013.

[ST18] Aaron Sidford and Kevin Tian. Coordinate methods for accelerating ℓ_∞ regression and faster approximate maximum flow. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018*, pages 922–933. IEEE Computer Society, 2018.

[ZO17] Zeyuan Allen Zhu and Lorenzo Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, volume 67 of *LIPIcs*, pages 3:1–3:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.