On Accelerated Methods in Optimization

Andre Wibisono  Ashia C. Wilson
wibisono@berkeley.edu  ashia@berkeley.edu

September 14, 2015

Abstract

In convex optimization, there is an acceleration phenomenon in which we can boost the convergence rate of certain gradient-based algorithms. We can observe this phenomenon in Nesterov’s accelerated gradient descent, accelerated mirror descent, and accelerated cubic-regularized Newton’s method, among others. In this paper, we show that the family of higher-order gradient methods in discrete time (generalizing gradient descent) corresponds to a family of first-order rescaled gradient flows in continuous time. On the other hand, the family of accelerated higher-order gradient methods (generalizing accelerated mirror descent) corresponds to a family of second-order differential equations in continuous time, each of which is the Euler-Lagrange equation of a family of Lagrangian functionals. We also study the exponential variant of the Nesterov Lagrangian, which corresponds to a generalization of Nesterov’s restart scheme and achieves a linear rate of convergence in discrete time. Finally, we show that the family of Lagrangians is closed under time dilation (an orbit under the action of speeding up time), which demonstrates the universality of this Lagrangian view of acceleration in optimization.

1 Introduction

In convex optimization, many discrete-time algorithms can be interpreted as discretizing a continuous-time curve converging to the optimal solution $f^*$ of the optimization problem:

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

For example, the classical gradient descent algorithm in discrete time ($k \in \{0, 1, 2, \ldots\}$)

$$x_{k+1} = x_k - \epsilon \nabla f(x_k) \quad (1.1)$$

can be viewed as the algorithm obtained by applying the backward-Euler method to discretize gradient flow ($t \geq 0$)

$$\dot{X}_t = -\nabla f(X_t). \quad (1.2)$$
Many methods, including (1.1) and (1.2) above, can be interpreted as minimizing a regularized approximation of the objective function \( f \). Indeed, gradient descent can be written as:

\[
x_{k+1} = x_k + v_k \\
v_k = \arg \min_v \left\{ f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{\epsilon} \cdot \frac{1}{2} \|v\|^2 \right\}
\]  

(1.3)

whereas gradient flow can be written as:

\[
\dot{X}_t = \arg \min_v \left\{ f(X_t) + \langle \nabla f(X_t), v \rangle + \frac{1}{2} \|v\|^2 \right\}.
\]  

(1.4)

Moreover, (1.3) and (1.4) have matching convergence rates; gradient descent has convergence rate

\[
f(x_k) - f^* \leq O \left( \frac{1}{\epsilon^k} \right)
\]  

(1.5)

when \( f \) is smooth (has \( \frac{1}{\epsilon} \)-Lipschitz gradients) and convex, where the \( O(\frac{1}{\epsilon^k}) \) term in the bound above is with respect to \( k \rightarrow \infty \), with \( \epsilon \) fixed (more precisely, \( f(x_k) - f^* \leq \frac{C}{\epsilon^k} \) for all sufficiently large \( k \), for some constant \( C > 0 \)). Similarly, gradient flow has convergence rate

\[
f(X_t) - f^* \leq O \left( \frac{1}{t} \right)
\]  

(1.6)

when \( f \) is convex, without requiring smoothness, and the \( O(\frac{1}{t}) \) term above is with respect to \( t \rightarrow \infty \). Note that the backward-Euler method discretizes the curve using the identification \( x_k = X_t \), \( x_{k+1} = X_{t+\delta} \approx X_t + \delta \dot{X}_t = x_k + v_k \), with the time-step \( \delta \) set equal to the step size \( \epsilon \) of the discrete time algorithm (equivalently, with time scaling \( t = \epsilon k \)).

### 1.1 Summary of Results

The link between discrete-time algorithms and continuous-time curves, and their matching properties (i.e. convergence rates) extends far beyond gradient descent (1.1) and gradient flow (1.2). We begin (Section 2) by studying higher-order gradient algorithms \( \mathcal{G}_p (p \geq 2) \):

\[
x_{k+1} = x_k + v_k \\
v_k = \arg \min_v \left\{ f_{p-1}(v; x_k) + \frac{1}{\epsilon} \cdot \frac{1}{p} \|v\|^p \right\},
\]  

(1.7)

where \( f_{p-1}(v; x) \) is the \((p-1)\)-st Taylor approximation of \( f(x + v) \) centered at \( x \):

\[
f_{p-1}(v, x) = \sum_{i=0}^{p-1} \frac{1}{i!} f(x_k) v^i = f(x_k) + \langle \nabla f(x_k), v \rangle + \cdots + \frac{1}{(p-1)!} \nabla^{p-1} f(x_k)[v, \ldots, v].
\]  

(1.8)

The \( p \)-th order gradient method \( \mathcal{G}_p \), with the ansatz \( x_k = X_t \), \( x_{k+1} = X_{t+\delta} \approx X_t + \delta \dot{X}_t = x_k + v_k \), and time-step \( \delta = \epsilon \frac{1}{p-1} \) (equivalently, with time scaling \( t = \epsilon \frac{1}{p-1} k \)), discretizes a \( p \)-th order rescaled gradient flow:

\[
\dot{X}_t = \arg \min_v \left\{ f(X_t) + \langle \nabla f(X_t), v \rangle + \frac{1}{p} \|v\|^p \right\}.
\]  

(1.9)
Note (1.9) can also be written as:

\[
\dot{X}_t = \frac{\nabla f(X_t)}{||\nabla f(X_t)||_{p^{-1}}}.
\]  

(1.10)

Furthermore, (1.7) and (1.9) have matching convergence rates; the \( p \)-th order gradient method has convergence rate:

\[
f(x_k) - f^* \leq O\left(\frac{1}{\epsilon k^{p-1}}\right)
\]

when \( \nabla^{(p-1)} f \) is \( \frac{(p-1)!}{\epsilon} \)-smooth and \( f \) is convex, and the rescaled gradient flow has convergence rate:

\[
f(X_t) - f^* \leq O\left(\frac{1}{t^{p-1}}\right)
\]

when \( f \) is convex.

In Section 3, we present an algorithm which generalizes accelerated gradient descent \([7]\) and the accelerated Newton method \([9]\), and accelerates the family of higher-order gradient algorithms (1.7). Building on the work of Su, Boyd, and Candes \([13]\) (for the \( p = 2 \) Euclidean case), in Section 3.3 we show that the \( p \)-th order accelerated gradient method \( \vec{G}_p \) discretizes a second-order differential equation we call Nesterov flow:

\[
\ddot{X}_t + \frac{p+1}{t} \dot{X}_t + C p^2 t^{p-2} \nabla^2 h \left( X_t + \frac{t}{p} \dot{X}_t \right)^{-1} \nabla f(X_t) = 0
\]

under the time step \( \delta = \frac{1}{t} \) (or time scaling \( t^p = \epsilon k^p \)). Moreover, the \( p \)-th order accelerated gradient algorithm \( \vec{G}_p \) and its corresponding Nesterov flow have matching convergence rates; the \( p \)-th order accelerated gradient method has convergence rate:

\[
f(x_k) - f^* \leq O\left(\frac{1}{\epsilon k^{p}}\right)
\]

when \( \nabla^{(p-1)} f \) is \( \frac{(p-1)!}{\epsilon} \)-smooth and \( f \) is convex, and the corresponding Nesterov flow has convergence rate (Section 4.1):

\[
f(X_t) - f^* \leq O\left(\frac{1}{t^{p}}\right)
\]

when \( f \) is convex. Note that the family of Nesterov flows are second-order differential equations in time and the rescaled gradient flows are first-order differential equations in time.

In Section 5, we show that the Nesterov flows are a subfamily of the Bregman flows:

\[
\ddot{X}_t + \gamma_t \dot{X}_t + e^{\beta_t} \nabla^2 h \left( X_t + e^{-\alpha_t} \dot{X}_t \right)^{-1} \nabla f(X_t) = 0
\]

where \( \alpha_t = - \log t + \log p \), \( \beta_t = (p-2) \log t + 2 \log p + \log C \), and \( \gamma_t = (p+1) \log t - \log p \). Under an ideal scaling relationship between \( \alpha_t, \beta_t, \gamma_t \) (satisfied by the Nesterov flows), each Bregman flow satisfies the Euler-Lagrange equation of a Bregman Lagrangian functional:

\[
\mathcal{L}_{\alpha, \beta, \gamma}(x, v, t) = e^{\alpha_t} D_h(x + e^{-\alpha_t} v, x) - e^{\beta_t} f(x).
\]  

(1.11)
Therefore, the family of Nesterov flows (4.1) can be interpreted as optimal curves under the principle of least action, which posits that curves evolve so as to minimize a quantity known as an action, defined as the time integral of a Lagrangian functional $\mathcal{L}(X, \dot{X}, t)$.

In Section 6.1, we introduce exp-Nesterov flows, another subfamily of the Bregman flows that satisfies the ideal scaling (where $\alpha_t = \log c$, $\beta_t = ct + 2 \log c$, $\gamma_t = ct - \log c$). The exp-Nesterov flows have an improved convergence rate:

$$f(X_t) - f^* \leq O\left(\frac{1}{e^{ct}}\right).$$

In Section 6.2, we show how to discretize the exp-Nesterov flow, and with the additional assumption of uniform convexity, obtain a discrete-time algorithm with matching linear rate. The algorithm presented generalizes the restart scheme of Nesterov [9], and is optimal (attains the lower bound [7, Section 2.2.1]) when $f$ is both smooth and strongly convex (i.e. $p = 2$).

Finally, in Section 7 we demonstrate how time can be used as an organizing tool to understand the various algorithms presented in this paper. Indeed, in continuous time optimization, if we start with a curve $X_t$ with convergence rate $f(X_t) - f^* \leq O(e^{\rho t})$, we can simply consider the sped-up curve $Y_t = X_{\tau(t)}$, where $\tau: \mathbb{R}_+ \rightarrow \mathbb{R}_+$ is a monotonically increasing function. This new curve $Y_t$ has faster convergence rate $f(Y_t) - f^* \leq O(e^{\rho \tau(t)})$, where $\rho_{\tau(t)} \geq \rho_t$. In this paper, we explore groups of time dilation functions $\tau$ and their corresponding group action on the space of curves. We show that the family of Bregman Lagrangian functionals forms an orbit under the group action of time dilation; moreover, the family of Nesterov Lagrangian functionals (Section 4) and the family of exp-Nesterov Lagrangian functionals (Section 6) form isomorphic sub-orbits. We can therefore interpret the curves corresponding to family of accelerated methods $\mathcal{G}$ as the result of speeding up (or traversing faster) the single curve corresponding to accelerated gradient descent. The cost for translating these faster curves into discrete-time algorithms (in addition to significant computational costs) is increasingly restrictive smoothness assumptions on the function.

We summarize in Figure 1 the relations between the objects we study in this paper. We see a consistent, almost parallel structure between continuous time (top layer) and discrete time (bottom layer). As the key conceptual message of this paper, we find there is a big difference in the nature of first-order equations (such as gradient flow) and second-order equations (such as accelerated gradient flow), due to the connection to the Lagrangian framework. Working with second-order equations provides better results in both continuous and discrete time.

1.2 Notation and Preliminaries

We formalize our setting and recall some basic definitions. Our objective is to minimize a convex objective function $f: \mathcal{X} \rightarrow \mathbb{R}$, which means the graph of $f$ lies above any tangent hyperplanes:

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$$

or equivalently, any intermediate value is at most the average value (Jensen’s inequality):

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$
Figure 1: Top layer is continuous time (first and second order equations), bottom layer is discrete time (first and second order algorithms).
for all \( x, y \in \mathcal{X} \) and \( 0 \leq \lambda \leq 1 \). We assume \( f \) is smooth and continuously differentiable as many times as necessary.

The domain \( \mathcal{X} \) is an open convex subset of a vector space, say \( \mathcal{X} \subseteq \mathbb{R}^d \) (we take \( \mathcal{X} = \mathbb{R}^d \) for simplicity). In particular, we take the point of view that \( \mathcal{X} \) is a manifold. In particular, any point \( x \in \mathcal{X} \) has a tangent space \( T_x \mathcal{X} \) (vector space of instantaneous displacements \( v \) such that \( x + \varepsilon v \in \mathcal{X} \) for small \( \varepsilon > 0 \); Since \( \mathcal{X} \subseteq \mathbb{R}^d \), we can identify \( T_x \mathcal{X} \) with \( \mathcal{X} \) (or \( \mathbb{R}^d \)) itself. But now we can have an interesting mixing of timescales between the points \( x \in \mathcal{X} \) and the vectors \( v \in T_x \mathcal{X} \), which are now “promoted” from \( \varepsilon \) to the standard timescale. Indeed, we will see in Section 4 that Nesterov’s acceleration technique involves the choice of mixing \( x \) and \( v \) using \( \varepsilon = \frac{t}{p} \)—which is counterintuitive because it is increasing, and yields sublinear rate of convergence. On the other hand, the exponential variant of Nesterov uses \( \varepsilon = \frac{1}{c} \)—which is more reasonable, and yields linear rate of convergence.

The cotangent space \( T^*_x \mathcal{X} \) is the dual vector space to \( T_x \mathcal{X} \) (the space of linear functional \( \phi \) on \( T_x \mathcal{X} \)). For example, the gradient \( \nabla f(x) \in T^*_x \mathcal{X} \) is a covector, which defines the directional derivative of \( f \):

\[
\langle \nabla f(x), v \rangle \equiv f'(x; v) := \lim_{\varepsilon \to 0} \frac{f(x + \varepsilon v) - f(x)}{\varepsilon}.
\]

(1.13)

Because we use the \( \ell_2 \)-norm, we can identify \( T_x \mathcal{X} \cong T^*_x \mathcal{X} \) by the identity map (which we implicitly use in (1.1)); but note that conceptually, they are different spaces. Similarly, the dual norm \( \| \cdot \|^* \) is also the same \( \ell_2 \)-norm, but we will maintain the distinction of \( \| \cdot \|_\ast \).

We say \( f \) is \( L \)-smooth of order \( \ell \geq 0 \) if \( \nabla^\ell f \) is \( L \)-Lipschitz (and \( \nabla^{(\ell+1)} \) is continuous):

\[
\| \nabla^\ell f(x) - \nabla^\ell f(y) \|_\ast \leq L \| x - y \|.
\]

(1.14)

The case \( \ell = 0 \) means \( f \) is Lipschitz, and \( \ell = 1 \) is the usual smoothness definition (\( \nabla f \) is Lipschitz). We say \( h \) is \( \sigma \)-uniformly convex of order \( p \geq 2 \) (\( p = 2 \) is strong convexity) if:

\[
D_h(y, x) \geq \frac{\sigma}{p} \| y - x \|^p.
\]

(1.15)

where \( D_h(y, x) = h(y) - h(x) - \langle \nabla h(x), y - x \rangle \geq 0 \) is the Bregman divergence induced by a strictly convex distance generating function \( h: \mathcal{X} \to \mathbb{R} \).

### 1.3 Gradient algorithms

We review gradient-based algorithms that correspond to first-order equations in continuous time.

#### 1.3.1 Mirror descent and natural gradient flow

The mirror descent algorithm

\[
x_{k+1} = \arg \min_v \left\{ \langle \nabla f(x_k), x \rangle + \frac{1}{\varepsilon} D_h(x, x_k) \right\}
\]

(1.16)
which can be written as

\[ x_{k+1} = x_k + v_k \]

\[ v_k = \arg\min_v \left\{ f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{\epsilon} D_h(x_k + v, x_k) \right\} \]

measures displacement by the Bregman divergence. Note that gradient descent (1.3) takes \( h(x) = \frac{1}{2} \|x\|^2 \), \( \mathcal{X} = \mathbb{R}^d \), and the classical multiplicative weight update uses \( h(x) = -\sum x_i \log x_i \), \( \mathcal{X} \) is the probability simplex. In general, there are often suitable choices of \( h \) that confer some computational gain in practice (e.g., milder dimension dependence). Similar to (1.5), mirror descent has convergence rate

\[ f(x_k) - f^* \leq O \left( \frac{1}{\epsilon k} \right) \] (1.17)

when \( \nabla f \) is \( \frac{1}{\epsilon} \)-Lipschitz, and \( h \) is strongly convex (i.e., uniformly convex of order 2).

In continuous time, mirror descent corresponds to (with \( \delta = \epsilon, t = \epsilon k \)) natural gradient flow:

\[ \dot{X}_t = -\nabla^2 h(X_t)^{-1} \nabla f(X_t) \] (1.18)

which can be cast as the solution to the optimization problem

\[ \dot{X}_t = \arg\min_v \left\{ f(X_t) + \langle \nabla f(X_t), v \rangle + \frac{1}{2} \| v \|^2_{h(X_t)} \right\} \] (1.19)

Natural gradient flow is a steepest descent direction when the metric in \( \mathcal{X} \) is induced by the Hessian \( \nabla^2 h \). Thus, mirror descent (1.16) can be seen as an alternative discretization of the natural gradient flow, another being Amari’s [2] natural gradient descent1:

\[ x_{k+1} = x_k - \epsilon \nabla^2 h(x_k)^{-1} \nabla f(x_k). \] (1.20)

Similar to (1.19), we can interpret (1.20) as the solution to the optimization problem

\[ x_{k+1} = x_k + v_k \]

\[ v_k = \arg\min_v \left\{ f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{\epsilon} \cdot \frac{1}{2} \| v \|^2_{h(x_k)} \right\}. \]

where \( \| v \|^2_{h(x)} = \langle \nabla^2 h(x) v, v \rangle \) is the Hessian norm induced by \( h \). Furthermore, like gradient flow (1.6), natural gradient flow has convergence rate

\[ f(X_t) - f^* \leq O \left( \frac{1}{t} \right) \] (1.21)

1This equivalence has also been observed by [12]; see Appendix A for further discussion.
1.3.2 Newton’s method and Newton’s flow

**Newton’s method** optimizes a quadratic approximation of the objective function $f$:

$$x_{k+1} = \arg \min_x \left\{ f(x_k) + \langle \nabla f(x_k), x \rangle + \frac{1}{2\epsilon} (\nabla^2 f(x_k)(x - x_k), x - x_k) \right\} \quad (1.22)$$

can be written explicitly as

$$x_{k+1} = x_k - \epsilon \nabla^2 f(x_k)^{-1} \nabla f(x_k). \quad (1.23)$$

The original Newton’s method corresponds to $\epsilon = 1$, but there have been many proposed choices of step size $\epsilon$ to improve stability and ensure convergence (e.g., see [10] and references therein).

We can also interpret (1.23) as natural gradient descent (1.20) when $h = f$. Thus, in continuous time it corresponds to *Newton’s flow*:

$$\dot{X}_t = -\nabla^2 f(X_t)^{-1} \nabla f(X_t) \quad (1.24)$$

which is natural gradient flow (1.18) with $f = h$. However, convergence results for the scheme (1.23) are difficult to obtain and have all been local. Only in special cases (e.g., self-concordance, a local Lipschitz condition on $\nabla^2 f$) are we able to have any strong guarantee on Newton’s method.

1.3.3 Cubic-regularized Newton’s method and rescaled gradient flow

To address this issue, Nesterov and Polyak [10] proposed *cubic-regularized Newton’s method*, which optimizes a second-order approximation of $f$ plus regularization:

$$x_{k+1} = \arg \min_x \left\{ f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2} (\nabla^2 f(x_k)(x - x_k), x - x_k) + \frac{1}{3\epsilon} \| x - x_k \|^3 \right\} \quad (1.25)$$

They showed [10, Theorem 4] that if $\nabla^2 f$ is $\frac{2}{\epsilon}$-Lipschitz, then (1.25) has convergence rate guarantee

$$f(x_k) - f^* \leq \frac{27\|x_0 - x^*\|^2}{\epsilon(k + 3)^2} = O\left( \frac{1}{\epsilon k^2} \right). \quad (1.26)$$

As mentioned in Section 1.1, we show (Section 2.2) that in continuous time (with $\delta = \epsilon^{1/2}$, $t^2 = \epsilon k^2$), (1.25) corresponds to the rescaled gradient flow:

$$\dot{X}_t = -\nabla f(X_t) / \| \nabla f(X_t) \|^{1/2} \quad (1.27)$$

and that (1.27) has matching convergence rate

$$f(X_t) - f^* \leq O\left( \frac{1}{t^2} \right). \quad (1.28)$$

Finally, we note that by adding a regularization term in Newton’s method (1.25), Nesterov and Polyak changed the problem from discretizing a Newton’s flow (1.24) to rescaled gradient flow (1.27). Thus, the two variants (1.23), (1.25) differ quite a bit in continuous time.

---

Note, Newton’s flow is explicitly solvable: $X_t = \nabla f^*(e^{-t} \nabla f(X_0))$, where $f^*$ is the convex dual of $f$. 

8
1.4 Accelerated gradient algorithms

We review accelerated gradient algorithms that correspond to second-order equations in continuous time.

1.4.1 Accelerated gradient descent

Nesterov’s accelerated gradient descent \([6, 7]\) improves the performance of gradient descent (1.1) from \(O\left(\frac{1}{\epsilon k}\right)\) to the optimal \(O\left(\frac{1}{\epsilon k^2}\right)\). This gain is achieved not by strengthening the assumption on \(f\), but by incorporating the displacement \(x_k - x_{k-1}\) to shift the point at which we query the gradient \(\nabla f\) (thus, this method is often referred to as gradient descent “with momentum”). The algorithm \([7, (2.2.6)]\) maintains three sequences \(\{x_k\}, \{y_k\}, \{z_k\}\) and proceeds as follows. For any \(y_0 = z_0 \in \mathcal{X}, k \geq 0\):

\[
\begin{align*}
x_k &= \tau_k z_k + (1 - \tau_k) y_k, \tag{1.29a} \\
y_{k+1} &= x_k - \epsilon \nabla f(x_k), \tag{1.29b} \\
z_{k+1} &= z_k - \frac{\epsilon}{\tau_k} \nabla f(x_k). \tag{1.29c}
\end{align*}
\]

Here \(\epsilon > 0\) is the step size, and \(\tau_k \in (0, 1)\) is defined recursively by \(\tau_{-1} = 1\) and the rule, for \(k \geq 0\):

\[
\frac{\tau^2_k}{1 - \tau_k} = \tau^2_{k-1}. \tag{1.30}
\]

We can also see that \(\tau_k = \frac{2}{k} + o(\frac{1}{k})\), for indeed we have \(\frac{\tau^2_k}{1 - \tau_k} = \frac{4/k^2}{1 - 2/k} = \frac{4}{k(k-2)} \approx \frac{4}{(k-1)^2} = \tau^2_{k-1}\). Nesterov showed \([7, \text{Theorem 2.2.2}]\) that when \(\nabla f\) is \(\frac{1}{\epsilon}\)-Lipschitz, then (1.29) satisfies:

\[
f(y_k) - f^* \leq \frac{4\|x_0 - x^*\|^2}{\epsilon (k + 2)^2} = O\left(\frac{1}{\epsilon k^2}\right), \tag{1.31}
\]

which improves the \(O(1/\epsilon k)\) rate (1.5) of gradient descent, and matches the lower bound.

As pointed out by Su, Boyd and Candes \([13]\), in continuous time accelerated gradient descent corresponds to the second order equation:

\[
\ddot{X}_t + \frac{3}{t} \dot{X}_t + \nabla f(X_t) = 0 \tag{1.32}
\]

with time scaling \(\delta = \frac{1}{\epsilon^2}, t^2 = \epsilon k^2\), and matching convergence rate \([13, \text{Theorem 3.2}]\):

\[
f(X_t) - f^* \leq O\left(\frac{1}{t^2}\right). \tag{1.33}
\]

1.4.2 Accelerated mirror descent

In \([8]\), Nesterov proposed \textit{accelerated mirror descent}, which proceeds much like the Euclidean case (1.29), except we replace the \(z\)-update (1.29c) by a (weighted) mirror descent step (1.16).
The algorithm \[8, (3.11)\] maintains three sequences \( \{x_k\}, \{y_k\}, \{z_k\} \) and proceeds as follows. For any \( x_0 \in \mathcal{X}, \ k \geq 0 \):

\[
y_k = \arg \min_y \left\{ f(x_k) + \langle \nabla f(x_k), y - x_k \rangle + \frac{1}{2\epsilon} \| y - x_k \|^2 \right\} \tag{1.34a}
\]

\[
z_k = \arg \min_z \left\{ \sum_{i=0}^{k} \frac{i+1}{2} \left[ f(x_i) + \langle \nabla f(x_i), z - x_i \rangle \right] + \frac{1}{\epsilon} \cdot \frac{1}{\sigma} D_h(z, x_0) \right\} \tag{1.34b}
\]

\[
x_{k+1} = \frac{2}{k+3} z_k + \frac{k+1}{k+3} y_k \tag{1.34c}
\]

Under the same assumption as mirror descent (\( \nabla f \) is \( \frac{1}{\epsilon} \)-smooth and \( h \) is \( \sigma \)-strongly convex), the accelerated version (1.34) has improved (optimal) convergence [8, Theorem 2]:

\[
f(y_k) - f^* \leq \frac{4D_h(x^*, x_0)}{\epsilon \sigma (k+1)(k+2)} = O \left( \frac{1}{\epsilon k^2} \right). \tag{1.35}
\]

Note, using the equivalence of mirror descent as the cascaded version of dual averaging, we can write the update (1.34b) above recursively using the standard mirror descent algorithm:\(^3\)

\[
z_k = \arg \min_z \left\{ \frac{k+1}{2} \langle \nabla f(x_k), z \rangle + \frac{1}{\epsilon} \cdot \frac{1}{\sigma} D_h(z, z_{k-1}) \right\}. \tag{1.36}
\]

In the Euclidean case (when \( h = \frac{1}{2} \| \cdot \|^2 \) and setting \( \sigma = 1 \)), the update (1.36) above simplifies to the explicit rule \( z_k = z_{k-1} - \frac{\epsilon (k+1)}{2} \nabla f(x_k) \), recovering accelerated gradient descent (1.29).

Similar to (1.37), accelerated mirror descent corresponds to the second order equation:

\[
\ddot{X}_t + \frac{3}{t} \dot{X}_t + \nabla^2 h \left( X_t + \frac{t}{2} \dot{X}_t \right)^{-1} \nabla f(X_t) = 0 \tag{1.37}
\]

with time scaling \( \delta = \frac{1}{\epsilon^2}, t^2 = \epsilon k^2 \), and matching convergence rate:

\[
f(X_t) - f^* \leq O \left( \frac{1}{t^2} \right). \tag{1.38}
\]

1.4.3 Accelerated cubic-regularized Newton’s method

In [9], Nesterov proposed the accelerated cubic-regularized Newton’s method, which proceeds as (1.29), except we replace the \( y \)-update (1.29b) by cubic-regularized Newton’s method (1.25). The algorithm [9, (4.8)] maintains three sequences \( \{x_k\}, \{y_k\}, \{z_k\} \) and proceeds as follows. For any \( x_0 \in \mathcal{X}, \ k \geq 0 \):

\[
y_k = \arg \min_y \left\{ f(x_k) + \langle \nabla f(x_k), y - x_k \rangle + \frac{1}{3\epsilon} \| y - x_k \|^3 \right\} \tag{1.39a}
\]

\[
z_k = \arg \min_z \left\{ \sum_{i=1}^{k} \frac{i(i+1)}{2} \left[ f(y_i) + \langle \nabla f(y_i), z - y_i \rangle \right] + \frac{2}{\epsilon} \| z - x_0 \|^3 \right\}, \tag{1.39b}
\]

\[
x_{k+1} = \frac{3}{k+3} z_k + \frac{k}{k+3} y_k. \tag{1.39c}
\]

\(^3\)This observation has also been reported by [1].
Under the same assumption as the cubic-regularized Newton’s method, $\nabla^2 f$ is $\frac{2}{\epsilon}$-Lipschitz, the accelerated algorithm (1.39) has convergence rate [9, Theorem 2]:

$$f(y_k) - f^* \leq \frac{14\|x_0 - x^\star\|^3}{\epsilon k(k + 1)(k + 2)} = O\left(\frac{1}{\epsilon k^3}\right).$$

(1.40)

As noted in Section 1.1, we show (Section 3.3) the accelerated algorithm (1.39) corresponds to the following differential equation:

$$\ddot{X}_t + \frac{4}{t} \dot{X}_t + 9t \nabla^2 h\left(X_t + \frac{t}{3} \dot{X}_t\right)^{-1} \nabla f(X_t) = 0$$

(1.41)

with $\delta = \epsilon^{\frac{1}{3}}, t^3 = \epsilon k^3$, where $h(x) = \frac{1}{3}\|x\|^3$ in (1.39). Furthermore, (1.41) has matching convergence rate:

$$f(X_t) - f^* \leq O\left(\frac{1}{t^3}\right).$$

(1.42)

2 Higher-order gradient methods and rescaled gradient flows

We study the family of higher-order gradient methods $G_p \in G$ in discrete time, which corresponds to first-order rescaled gradient flow in continuous time, with time step $\delta = \epsilon^{\frac{1}{p-1}}$, and matching convergence rate $O(1/t^{p-1}) = O(1/\epsilon k^{p-1})$.

Surrogate optimization. Recall in discrete time, many optimization algorithms proceed by minimizing a surrogate function:

$$x_{k+1} = \arg\min_{x \in \mathcal{X}} g(x; x_k).$$

(2.1)

Here $g(x; x_k)$ is a surrogate function that majorizes the objective function $f(x)$, which means for any reference point $x_k \in \mathcal{X}$ and for any point $x \in \mathcal{X}$, it satisfies the inequality:

$$g(x; x_k) \geq f(x)$$

(2.2)

and equality holds at $x = x_k$. The property (2.2) above implies the algorithm (2.1) is a descent method, which means the function value $f(x_k)$ decreases along the iteration of the algorithm:

$$f(x_{k+1}) \overset{(2.2)}{\leq} g(x_{k+1}; x_k) \overset{(2.1)}{\leq} g(x_k; x_k) = f(x_k).$$

(2.3)

We can minimize $f$ by finding an appropriate surrogate function $g(x; x_k)$ that is more tractable to minimize, and performing the descent algorithm (2.1). Moreover, under various assumptions, we can quantify the decrease in the function value (2.3), resulting in a rate of convergence for the algorithm (2.1).

\[4\] See [5] and the references therein for more information on majorization-minimization principle in optimization.
**Surrogate via Taylor expansion.** A natural technique for constructing a surrogate function $g(x; y)$ is to use the Taylor approximation of $f$ at $x$ from the reference point $y$

$$f_{p-1}(x; y) = \sum_{i=0}^{p-1} \frac{1}{i!} \nabla^i f(y)(x-y)^i = f(y) + \langle \nabla f(y), x-y \rangle + \cdots + \frac{1}{(p-1)!} \nabla^{p-1} f(y)(x-y)^{p-1}. \tag{2.4}$$

If $f$ is $L$-Lipschitz of order $p-1$ (1.14), then we have a bound on the approximation error of $f_{p-1}$:

$$|f(x) - f_{p-1}(x; y)| \leq \frac{L}{p!} \|x - y\|^p. \tag{2.5}$$

Therefore, we have the family of (regularized) **Taylor surrogate functions**, for $p \geq 1$:

$$g_p(x; y) := f_{p-1}(x; y) + \frac{L}{p!} \|x - y\|^p. \tag{2.6}$$

Note also the tangency property $g_p(x; x) = f_{p-1}(x; x) = f(x)$, so $g_p$ is indeed a surrogate function.

### 2.1 Higher-order gradient method

The Taylor surrogate functions (2.6) give rise to **higher-order gradient methods** $G_p \in \mathcal{G}$, defined by the update equation:

$$x_{k+1} = \arg\min_x \left\{ f_{p-1}(x; x_k) + \frac{1}{\epsilon} \cdot \frac{1}{p} \|x - x_k\|^p \right\}. \tag{2.7}$$

The $p = 2$ case gives the gradient descent algorithm (1.1), and $p = 3$ is Nesterov and Polyak’s [10] cubic-regularized Newton’s method (1.25). Note also that $p = 1$ gives a constant sequence, so we only consider $p \geq 2$. In (2.7), we write the Lipschitz constant $L = \frac{(p-1)!}{\epsilon} = O(\frac{1}{\epsilon})$ in terms of the step size $\epsilon > 0$, representing the discretization parameter of the algorithm.\(^5\) Our discussion above says that if $f$ is $\frac{(p-1)!}{\epsilon}$-smooth of order $p-1$, then the $p$-th gradient algorithm $G_p$ (2.7) is a descent method, since $f(x_{k+1}) \leq f(x_k)$.

Moreover, we can use the convexity structure of $f$ to further ensure a quantitative decrease in the residual $\delta_k = f(x_k) - f^* \geq 0$:

**Lemma 1.** If $f$ is convex and $\frac{(p-1)!}{\epsilon}$-smooth of order $p-1$, then the following holds for (2.7):

$$\delta_{k+1} \leq \delta_k - \frac{1}{R^{p-1}} \cdot \delta_k^{\frac{p}{p-1}}. \tag{2.8}$$

where $R = \sup_{f(x) \leq f(x_0)} \|x - x^*\|$ is the radius of the level set from the initial point $x_0$.

The proof of Lemma 1 is in Appendix B.1. Using the (**discrete time**) energy functional $\mathcal{E}_k = \delta_k^{\frac{p}{p-1}}$, we obtain following convergence rate for $G_p$, generalizing [9, Theorem 1]:

\(^5\)This assumption vanishes in continuous time, since $\frac{1}{\epsilon} \to \infty$ as $\epsilon \to 0$. 


Theorem 2. If $f$ is convex and $(p-1)!\varepsilon$-smooth of order $p-1$, then the following holds for (2.7):

$$f(x_k) - f^* \leq \frac{(p-1)^{p-1}Rp}{ek^{p-1}} = O\left(\frac{1}{ek^{p-1}}\right).$$

(2.9)

Thus, the family of higher-order gradient methods $G_p \in \mathcal{G}$ in discrete time has a nice sequential structure. In particular, there is a consistent pattern whereby as $p$ progresses in $\{2,3,\ldots\}$, the polynomial convergence rate $O(1/ek^{p-1})$ decreases, but at the cost of increasingly strict $(p-1)$-st order smoothness assumption on $f$. This suggests there is a fundamental tradeoff in discrete time between speed of convergence and strength of hypothesis required.

2.2 Rescaled gradient flow

In continuous time, gradient flow (1.2) is a member ($p = 2$) of a family of rescaled gradient flows:

$$\dot{X}_t = -\nabla f(X_t) \frac{1}{\|\nabla f(X_t)\|^{\frac{1}{p-1}}}.$$ 

(2.10)

When $\nabla f(X_t) = 0$, we define the right hand side of (2.10) to be 0. Note that (2.10) implies that the magnitude of the velocity at some time is proportional to (some power of) the gradient at that point:

$$\|\dot{X}_t\| = \|\nabla f(X_t)\|^{\frac{1}{p-1}}.$$ 

(2.11)

so we can also equivalently write the rescaled gradient flow as

$$\|\dot{X}_t\|^{p-2}\dot{X}_t = -\nabla f(X_t).$$ 

(2.12)

Furthermore, (2.12) is the optimality condition for the following optimization problem:

$$\dot{X}_t = \arg\min_v \left\{ f(X_t) + \langle \nabla f(X_t), v \rangle + \frac{1}{p} \|v\|^{p} \right\}.$$ 

(2.13)

Thus, we can view the rescaled gradient flow (2.10) as a generalization of gradient flow that replaces the squared norm in the optimization interpretation (1.4) by the $p$-th power of the norm, for $p \geq 2$.

If $X_t$ is a curve that evolves following the rescaled gradient flow (2.10), then, using the convexity of $f$, the energy functional:

$$\mathcal{E}_t = (f(X_t) - f^*)^{-\frac{1}{p-1}}$$ 

(2.14)

increases linearly over time ($\dot{\mathcal{E}}_t \geq \frac{1}{p-1}R^{-\frac{1}{p-1}}$, see Appendix B.2 for details), which implies the following rate of convergence:

Theorem 3. If $f$ is convex with bounded level sets, then the following holds for (2.10):

$$f(X_t) - f^* \leq \frac{(p-1)^{p-1}Rp}{t^{p-1}} = O\left(\frac{1}{t^{p-1}}\right)$$ 

(2.15)

The result (2.15) above matches the discrete time convergence rate (2.9) of the $p$-th order gradient method $G_p$. Moreover, notice we use the same energy functional (2.14) as in discrete time.
2.3 The relation between $\epsilon$ and $\delta$

The matching convergence rates (2.9), (2.15) suggest the following time scaling relation between continuous time $t \geq 0$ and discrete time $k \in \{2, 3, \ldots\}$:

$$t = \epsilon^{\frac{1}{p-1}} k. \quad (2.16)$$

So in one step of discrete time $k \mapsto k + 1$, the continuous time $t \equiv t_k$ increments by:

$$\delta \equiv \frac{dt}{dk} = \frac{t_{k+1} - t_k}{(k+1) - k} = \epsilon^{\frac{1}{p-1}}. \quad (2.17)$$

The scaling above suggests the interpretation of the $p$-th gradient algorithm $G_p$ as a discretization of the rescaled gradient flow with time step $\delta = \epsilon^{\frac{1}{p-1}}$.

We can understand this scaling phenomenon more explicitly by starting from the continuous time view. Suppose we have a continuous time curve $X_t$ (such as the rescaled gradient flow (2.10)), and we want to discretize it with time step $\delta > 0$. This means we build a discrete time sequence $x_k$ obtained by taking a snapshot of $X_t$ every $\delta$ increment of time, namely:

$$X_t = x_k \quad \Rightarrow \quad X_{t+\delta} = x_{k+1}. \quad (2.18)$$

Notice, to go from $x_k$ to $x_{k+1}$ in (2.18) above, we have to first let $X_t$ evolve in continuous time to $X_{t+\delta}$, which we then use as the value for $x_{k+1}$.

However, to discretize is to build a discrete time algorithm, which means we can only define $x_{k+1}$ in terms of the previous discrete iterates $x_k, x_{k-1}, \ldots$, without invoking the continuous time curve $X_t$. Thus, we have to approximate the continuous time evolution from $X_t$ to $X_{t+\delta}$.

For the first-order rescaled gradient flow (2.10), we approximate $X_{t+\delta}$ using a linear approximation:

$$X_{t+\delta} \approx X_t + \delta \dot{X}_t \quad (2.19)$$

with an error of order $o(\delta)$. Equivalently, we replace $\dot{X}_t$ in (2.13) by the discrete time difference:

$$\frac{x_{k+1} - x_k}{\delta} = \arg \min_v \left\{ \langle \nabla f(x_k), v \rangle + \frac{1}{p} \|v\|^p \right\}. \quad (2.20)$$

Or, writing $v = \frac{x_{k+1} - x_k}{\delta}$, the above can be written as:

$$x_{k+1} = \arg \min_x \left\{ f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{\delta^{p-1}} \cdot \frac{1}{p} \|x - x_k\|^p \right\}. \quad (2.21)$$

Thus, we obtain an algorithm similar to the $p$-th gradient method (2.13), where the step size $\epsilon$ is given by $\delta^{p-1}$, consistent with the time scaling (2.17). We can interpret the $p$-th gradient method (2.7) as a particular discretization technique of the rescaled gradient flow (2.10), which replaces the first-order approximation of $f$ in the “naive” discretization (2.21) by the $(p-1)$-st order approximation (2.7). By doing so, as well as assuming $(p-1)$-st order smoothness of $f$, the resulting discrete time algorithm $G_p$ has a $O(1/\epsilon k^{p-1})$ convergence rate which matches the $O(1/t^{p-1})$ bound in continuous time.

---

*We can also start by replacing $\dot{X}_t$ in (2.10) by (2.19) and achieve the same conclusion. Namely, (2.10) becomes $\frac{1}{\delta} (x_{k+1} - x_k) = -\nabla f(x_k)/\|\nabla f(x_k)\|^{\frac{p-2}{2}},$ or equivalently, $\|x_{k+1} - x_k\|^{p-2}(x_{k+1} - x_k) = -\delta^{p-1} \nabla f(x_k),$ which is (2.21).*
3 Accelerated higher-order gradient methods

In Section 1.4, we see the simple pattern in Nesterov’s constructions of accelerated methods [7, 8, 9]:

To accelerate an algorithm, couple it with a (suitably weighted) mirror descent step.

In this section, we extend Nesterov’s technique to accelerate all higher-order gradient methods. The accelerated $p$-th order gradient method $\vec{G}_p$ is obtained by coupling $G_p$ with a mirror descent step weighted by a polynomial of order $p - 1$. The accelerated algorithm $\vec{G}_p$ has an improved convergence rate $O(1/\epsilon k^p)$ under the same $(p - 1)$-st order smoothness assumption as $G_p$. Thus, just like $G_p$, the family of accelerated gradient methods $\vec{G}_p \in \vec{G}$ still maintains the nice sequential property of the polynomially decreasing convergence rates.

Throughout this section, we fix an integer $p \geq 2$, and assume $f$ is $(p - 1)!\epsilon$-smooth of order $p - 1$ (1.14). For the mirror descent step, we assume the distance generating function $h$ is $\sigma$-strongly convex of order $p$ (1.15), where $\sigma > 0$ is a constant (we can normalize $\sigma = 1$). For example, we can take $h$ to be the $p$-th power of the norm:

$$d_p(x) = \frac{1}{p}||x - x_0||^p$$

(for arbitrary reference point $x_0 \in \mathcal{X}$), which is $(\frac{1}{2})^{p-2}$-uniformly convex of order $p$ [9, Lemma 4].

3.1 Accelerated $p$-th order gradient method

The accelerated $p$-th order gradient method $\vec{G}_p$ maintains three sequences $\{x_k\}, \{y_k\}, \{z_k\}$ as follows. Starting from any $x_0 \in \mathcal{X}$, $k \geq 0$ the algorithm proceeds:

$$y_k = \arg \min_y \left\{ f_{p-1}(y; x_k) + \frac{2}{\epsilon} \cdot \frac{1}{p}||y - x_k||^p \right\}$$

$$z_k = \arg \min_z \left\{ \sum_{i=0}^{k} C p i^{(p-1)} \left[ f(y_i) + \langle \nabla f(y_i), z - y_i \rangle \right] + \frac{1}{\epsilon} \cdot \frac{1}{\sigma} D_h(z, x_0) \right\}$$

$$x_{k+1} = \frac{p}{k + p} z_k + \frac{k}{k + p} y_k$$

where $i^{(p-1)} := i(i + 1)\cdots(i + p - 2)$ denotes the rising factorial, and $C \leq (4p)^{-p}$ is a constant.

Note, the $y$-update (3.2a) above is the $p$-th gradient method, but with slightly larger regularization ($\frac{2}{\epsilon}$ with any $c > 1$; above is $c = 2$). As noted in Section 1.4, the $z$-update (3.2b) above is given in a “dual averaging” form, because (for example, when $\mathcal{X} = \mathbb{R}^d$) we can write it explicitly:

$$\nabla h(z_k) = \nabla h(x_0) - \epsilon \sigma C p \sum_{i=0}^{k} i^{(p-1)} \nabla f(y_i) = \nabla h(z_{k-1}) - \epsilon \sigma C pk^{(p-1)} \nabla f(y_k).$$

While preparing this paper, we became aware of an unpublished manuscript by Baes [3] who extended Nesterov’s technique of estimate sequence and constructed higher-order variants of the accelerated methods, essentially identical to ours. We nevertheless present our generalization of Nesterov’s proof in order to highlight the basic structure.
Therefore, the $z$-update (3.2b) can also be equivalently written recursively as a mirror descent step:

$$z_k = \arg \min_z \left\{ Cp_k^{(p-1)} \langle \nabla f(y_k), z \rangle + \frac{1}{\epsilon \sigma} D_h(z, z_{k-1}) \right\}. \quad (3.4)$$

But it turns out that the expanded form of the update (3.2b) is more convenient for us.

### 3.2 Convergence analysis

We can justify the performance of the accelerated algorithm (3.2) by following a straightforward generalization of Nesterov’s arguments [8, 9], which proceeds as follows. We first recall the following property for the $p$-th order gradient step in the $y$-update (3.2a). This lemma generalizes [9, Lemma 6], and its proof is provided in Appendix B.3.

**Lemma 4.** If $f$ is $(p-1)!\frac{1}{\epsilon}$-smooth of order $p-1$, then the $y$-update (3.2a) has the guarantee:

$$\langle \nabla f(y_k), x_k - y_k \rangle \geq \frac{1}{4} \frac{1}{\epsilon^{p-1}} \| \nabla f(y_k) \|^p. \quad (3.5)$$

With Lemma 4 in hand, we can proceed as follows. Let $\psi_k$ (Nesterov’s “estimate function”) denote the objective function in the $z$-update (3.2b):

$$\psi_k(x) = Cp \sum_{i=0}^{k} i^{(p-1)} [f(y_i) + \langle \nabla f(y_i), x - y_i \rangle] + \frac{1}{\epsilon \sigma} D_h(x, x_0). \quad (3.6)$$

Since $f$ is convex, each term in the summation above is at most $Cp i^{(p-1)} f(x)$. Noting that $\sum_{i=0}^{k} i^{(p-1)} = k^{(p)} / p$, this yields the following upper bound on the estimate function, for all $x \in \mathcal{X}$:

$$\psi_k(x) \leq Ck^{(p)} f(x) + \frac{1}{\epsilon \sigma} D_h(x, x_0). \quad (3.7)$$

Furthermore, the updates in (3.2) are constructed in such a way that we also have the following guarantee.

**Proposition 5.** If $C \leq (4p)^{-p}$, then for all $k \geq 0$:

$$Ck^{(p)} f(y_k) \leq \psi_k^* := \min_x \psi_k(x). \quad (3.8)$$

**Proof.** We proceed via induction on $k \geq 0$. The base case $k = 0$ is trivial since both sides equal 0. Now assume (3.8) holds for some $k \geq 0$; we will show it also holds for $k + 1$.

Since $h$ is $\sigma$-uniformly convex of order $p$, the rescaled Bregman divergence $\frac{1}{\epsilon \sigma} D_h(x, x_0)$ is $\frac{1}{\epsilon}$-uniformly convex. Thus, the estimate function $\psi_k$ (3.6) is also $\frac{1}{\epsilon}$-uniformly convex of order $p$. Since $z_k$ is the minimizer of $\psi_k$, this implies $\psi_k(x) \geq \psi_k^* + \frac{1}{\epsilon p} \| x - z_k \|^p$. We then apply the inductive hypothesis (3.8) and use the convexity of $f$ (1.12) to obtain the bound, for all $x \in \mathcal{X}$:

$$\psi_k(x) \geq Ck^{(p)} [f(y_{k+1}) + \langle \nabla f(y_{k+1}), y_k - y_{k+1} \rangle] + \frac{1}{\epsilon p} \| x - z_k \|^p. \quad (3.9)$$
Then by adding the \((k + 1)\)-st term in the definition of \(\psi_{k+1}\) (3.6) to both sides of (3.9), we obtain:

\[
\psi_{k+1}(x) \geq C(k + 1)^p \left[ f(y_{k+1}) + \langle \nabla f(y_{k+1}), x_{k+1} - y_{k+1} + \tau_k(x - z_k) \rangle \right] + \frac{1}{\epsilon^p} \|x - z_k\|^p \tag{3.10}
\]

where \(\tau_k = \frac{p(k+1)^{(p-1)}}{(k+1)^p} = \frac{p}{k+1}\), and in the above we have also used the definition of \(x_{k+1}\) as a convex combination of \(y_k\) and \(z_k\) with weight \(\tau_k\) (3.2c).

Note, the first term in (3.10) above gives our desired inequality (3.8) for \(k + 1\). So to finish the proof, we have to prove the remaining terms in (3.10) are nonnegative. We do so by applying two inequalities: We apply Lemma 4 to the term \(\langle \nabla f(y_{k+1}), x_{k+1} - y_{k+1} \rangle\). We also apply the classical Fenchel-Young inequality (e.g., [9, Lemma 2]): \(\langle s, h \rangle + \frac{1}{p} \|h\|^p \geq -\frac{p}{p-1} \|s\|^{\frac{p}{p-1}}\) with the choices \(h = \epsilon^{-\frac{1}{p}}(x - z_k)\) and \(s = \epsilon^{\frac{1}{p}}Cp(k + 1)^{(p-1)}\nabla f(y_{k+1})\). Then from (3.10), we obtain:

\[
\psi_{k+1}(x) \geq C(k + 1)^p \left[ f(y_{k+1}) + \left( 1 - \frac{p}{p-1} \frac{\frac{2p-1}{p^{p-1}}}{\epsilon^{p-1}} \right) \left\{ (k+1)^{(p-1)} \frac{p}{p-1} \right\} \nabla f(y_{k+1}) \right] \geq \|\nabla f(y_{k+1})\|^{\frac{p}{p-1}}. \tag{3.11}
\]

Notice that \((k+1)^{(p-1)}\) is less than \((k+1)^p\). Then from the assumption \(C \leq (4p)^{-p}\), we see that the second term inside the parenthesis above is nonnegative. Hence we conclude the desired inequality \(\psi_{k+1}(x) \geq C(k + 1)^p f(y_{k+1})\), finishing the induction.

Finally, we can combine the result of Proposition 5 with the basic estimate (3.7) at \(x = x^\ast\), to conclude a convergence rate on \(\mathcal{G}_p\), which we summarize in the following theorem.

**Theorem 6.** If \(f\) is \((p-1)!\)-smooth of order \(p - 1\), \(h\) is \(\sigma\)-uniformly convex of order \(p\), and \(C \leq (4p)^{-p}\), then the \(p\)-th order accelerated gradient algorithm (3.2) has convergence rate:

\[
f(y_k) - f^\ast \leq \frac{D_h(x^\ast, x_0)}{C \epsilon \sigma k^p} = O \left( \frac{1}{\epsilon k^p} \right).\tag{3.11}
\]

The result above shows that by plugging in the \(p\)-th gradient method \(\mathcal{G}_p\) into the accelerated algorithm (3.2), we boost its convergence rate from \(O(1/\epsilon k^{p-1})\) to \(O(1/\epsilon k^p)\). In particular, the family of accelerated gradient algorithms \(\mathfrak{G}\) still maintains the nice sequential pattern of polynomial convergence rates, just like \(\mathfrak{G}\).

### 3.3 Continuous time limit

We show that the continuous time limit \((\epsilon \to 0)\) of the accelerated algorithm \(\mathcal{G}_p\) (3.2) is a second order differential equation (with time scaling \(\delta = \epsilon^p\)). This is in contrast to the original \(p\)-th order gradient method \(\mathcal{G}_p\) (2.7), which corresponds to the first-order rescaled gradient flow in continuous time (with time scaling \(\delta = \epsilon^{p-1}\)). Here we sketch the transition from discrete to continuous time, and in Section 4.2 we will see the other view starting from the continuous time perspective.

We first note the following difference between the continuous time considerations of \(\mathcal{G}_p\) and \(\mathcal{G}_p\). The \(p\)-th gradient method \(\mathcal{G}_p\) is an algorithm that updates the sequence \(x_k\) by:

\[
x_{k+1} = A_\epsilon(x_k) \tag{3.12}
\]
where $A:\mathcal{X} \to \mathcal{X}$ is the operator that returns the minimizer of the optimization problem (2.7). The update (3.12) above is equivalent to modeling the (discrete time) velocity $v_k = x_{k+1} - x_k$ as a function of the current position, $v_k = A'(x_k) (= A(x_k) - x_k)$. As $\epsilon \to 0$, and by identifying $x_{k+1} = X_{t+\delta}$, $x_k = X_t$ with $\delta = \epsilon^{p-1}$, we recover a first order differential equation, the rescaled gradient flow (2.10).

On the other hand, the accelerated gradient algorithm $\vec{G}_p$ maintains three sequences:

$$
(x_{k+1}, y_{k+1}, z_{k+1}) = A_{k,\epsilon}(x_k, y_k, z_k)
$$

(3.13)

where the operator $A_{k,\epsilon}$ now changes over (discrete) time $k$. As in the preceding paragraph, as $\epsilon \to 0$ the update (3.13) above gives rise to a system of first order differential equations in the variables $X_t, Z_t$ (and $Y_t = X_t$), which is equivalent to a second order equation in $X_t$. Moreover, since $A_{k,\epsilon}$ depends with $k$, this second order equation has time-varying coefficients.

**Derivation.** We now analyze the updates (3.2) in the limit $\epsilon \to 0$, starting with the $z$-update (3.2b).

- **$z$-update.** As noted in Section 3.1, we can write the $z$-update recursively as:

$$
w_k = w_{k-1} - C\epsilon k^{(p-1)} \nabla f(y_k)
$$

(3.14)

where $w_k = \nabla h(z_k)$, and here we set $\sigma$ (the uniform convexity constant of $h$) to be 1 for simplicity. Now we invoke the hypothesis that the sequences $y_k, w_k$ are discrete time snapshots of continuous time curves $Y_t, W_t$ (similarly for $x_k, z_k$ with respect to $X_t, Z_t$) at each time increment $\delta > 0$. Specifically, we identify $y_k = Y_t, w_k = W_t$, and $w_{k-1} = W_{t-\delta}$, and use the linear approximation $W_{t-\delta} = W_t - \delta \dot{W}_t + o(\delta)$. Under these identifications, (3.14) becomes:

$$
\dot{W}_t = -\frac{\epsilon}{\delta} C\epsilon k^{(p-1)} \nabla f(Y_t) + o(1).
$$

(3.15)

With time increment $\delta$ we have the correspondence $t = \delta k$ or $k = t/\delta$, so $k^{(p-1)} \approx k^{p-1} = t^{p-1}/\delta^{p-1}$. Thus, on the right hand side of (3.15) we have the factor $\epsilon/\delta^p$. As $\epsilon \to 0$ and $\delta \to 0$, for the expression (3.15) above to have a meaningful content, we need to have the two variables to scale as $\epsilon = \delta^p$ (or $\epsilon = \Theta(\delta^p)$ in general). Under this scaling, (3.15) yields the differential equation for $W_t$:

$$
\dot{W}_t = -C t^{p-1} \nabla f(Y_t).
$$

(3.16)

Since $w_k = \nabla h(z_k)$, $W_t = \nabla h(Z_t)$, we also have $\frac{d}{dt} \nabla h(Z_t) = -C t^{p-1} \nabla f(Y_t)$, or equivalently:

$$
\dot{Z}_t = -C t^{p-1} \nabla^2 h(Z_t)^{-1} \nabla f(Y_t).
$$

(3.17)

Notice, (3.17) is a first order equation in time since it involves the velocity $\dot{Z}_t$, but second order in space since it involves the Hessian $\nabla^2 h(Z_t)$. 

18
• **y-update.** Observe, the y-update (3.2a) operates on a smaller time scale $\epsilon^{\frac{1}{p-1}}$. That is, from the first order optimality condition of $y_k$ (B.8), and the bound implied by the $\frac{(p-1)!}{\epsilon}$-Lipschitz assumption of $\nabla^{p-1} f$ (B.9), we have:

$$\|x_k - y_k\| \leq \epsilon^{\frac{1}{p-1}} \|\nabla f(y_k)\|^{\frac{1}{p-1}}. \quad (3.18)$$

With the identification $x_k = X_t$, $y_k = Y_t$, the bound (3.18) above shows the difference $X_t - Y_t = O(\epsilon^{\frac{1}{p-1}})$ is smaller than our time step $\delta = \epsilon^\frac{1}{p}$ (needed for (3.17)). Therefore:

$$X_t = Y_t. \quad (3.19)$$

• **x-update.** The x-update (3.2c) can be written as $x_{k+1} - x_k = \frac{p}{k+p}(z_k - x_k) + \frac{k}{k+p}(y_k - x_k)$. Identifying $x_{k+1} - x_k = \delta \dot{X}_t$, $z_k = Z_t$, and $\delta(k+p) = t$, as well as using the bound (3.18) with $\delta = \epsilon^\frac{1}{p}$, we get $\|\dot{X}_t - \frac{\rho}{p}(Z_t - X_t)\| \leq \frac{1}{\delta} \cdot \epsilon^{\frac{1}{p-1}} \|\nabla f(Y_t)\|^{\frac{1}{p-1}} \to 0$, which means:

$$\dot{X}_t = \frac{p}{t}(Z_t - X_t). \quad (3.20)$$

Thus, we conclude that the continuous time limit of the accelerated gradient method $\tilde{G}_p$ (3.2) is the system of first order differential equations (3.17), (3.19), (3.20). This system is equivalent to the following second order differential equation for $X_t$:

$$\ddot{X}_t + \frac{p+1}{t} \dot{X}_t + C p^2 t^{p-2} \nabla^2 h \left( X_t + \frac{t}{p} \dot{X}_t \right)^{-1} \nabla f(X_t) = 0. \quad (3.21)$$

Note, even though the accelerated gradient methods $\tilde{G}_p$ use higher-order derivatives of $f$, in continuous time they are all second order differential equations (3.21). This is in parallel to—but also in contrast from—how the $p$-th gradient method $G_p$ is a $(p - 1)$-st order algorithm (in space) but corresponds to a first order differential equation (in time), the rescaled gradient flow (2.10).

### 4 Nesterov flow

In this section we study the family of second order differential equation (3.21), the Nesterov flow:

$$\ddot{X}_t + \frac{p+1}{t} \dot{X}_t + C p^2 t^{p-2} \nabla^2 h \left( X_t + \frac{t}{p} \dot{X}_t \right)^{-1} \nabla f(X_t) = 0 \quad (4.1)$$

where $p > 0$ is not necessarily an integer, and $C > 0$ is a constant. Here we assume $f$ is convex and continuously differentiable, and $h$ is strictly convex so $\nabla^2 h$ is invertible. But we make no smoothness assumption on $f$ nor uniform convexity assumption on $h$.

---

8Lemma 4 gives the reverse $\|x_k - y_k\| \geq \frac{1}{\delta} \epsilon^{\frac{1}{p-1}} \|\nabla f(y_k)\|^{\frac{1}{p-1}}$, so the bound is tight. Indeed, like in Section 2.3, the $y$-update (3.2a) is a discretized rescaled gradient flow: $y_k = x_k - 2^{\frac{1}{p-1}} \epsilon^{\frac{1}{p-1}} \nabla f(x_k)/\|\nabla f(x_k)\|^{\frac{1}{p-1}} + o(\epsilon^{\frac{1}{p-1}})$. 
Notice, we can equivalently write Nesterov flow (4.1) as follows:

$$\frac{d}{dt} \nabla h \left( X_t + \frac{t}{p} \dot{X}_t \right) = \nabla^2 h \left( X_t + \frac{t}{p} \dot{X}_t \right) \left( \frac{p+1}{p} \dot{X}_t + \frac{t}{p} \ddot{X}_t \right) = -Cpt^{p-1} \nabla f(X_t).$$  \hfill (4.2)

As shown in Section 3.3, equation (4.2) is the continuous time limit of the $p$-th accelerated gradient method $\mathcal{G}_p$ (3.2), for integer $p \geq 2$. The rewriting (4.2) is nicer than (4.1) because it avoids the potential singularity problem at $t = 0$ (i.e., no term $\frac{t+1}{t}$). Indeed, by setting $t \to 0$ in (4.2) we see that if $p > 1$, then $\dot{X}_0 = 0$. That is, any Nesterov curve $X_t$ that evolves following (4.1), (4.2) must start from being at rest; it is the acceleration $\ddot{X}_t$ that drives the trajectory.

### 4.1 Convergence rate via energy functional

Our interest in Nesterov flow (4.1) stems from it being the continuous time limit of the $p$-th accelerated gradient method $\mathcal{G}_p$, which has convergence rate $O(1/\epsilon k^p)$. (Theorem 6). We now show Nesterov flow preserves this convergence rate, without any additional assumptions on $f, h$ beyond convexity. Define the energy functional:

$$\mathcal{E}_t = C t^p (f(X_t) - f^*) + D_h \left( x^*, X_t + \frac{t}{p} \dot{X}_t \right)$$  \hfill (4.3)

where recall, $x^* = \arg \min_x f(x)$ and $f^* = f(x^*)$. It has time derivative:

$$\dot{\mathcal{E}}_t = C t^{p-1} \left( f(X_t) - f^* + \frac{t}{p} \langle \nabla f(X_t), \dot{X}_t \rangle \right) - \left( \frac{d}{dt} \nabla h \left( X_t + \frac{t}{p} \dot{X}_t \right), x^* - X_t - \frac{t}{p} \dot{X}_t \right).$$  \hfill (4.4)

If $X_t$ is governed by Nesterov flow (4.1), then $\dot{\mathcal{E}}_t$ above simplifies to:

$$\dot{\mathcal{E}}_t = C t^{p-1} \left( f(X_t) - f^* + \langle \nabla f(X_t), x^* - X_t \rangle \right) \leq 0$$  \hfill (4.5)

where the last inequality follows from the convexity of $f$. This means energy is decreasing over time: $\mathcal{E}_t \leq \mathcal{E}_0 = D_h(x^*, X_0)$, for all $t \geq 0$. Since $D_h(x^*, X_t + \frac{t}{p} \dot{X}_t) \geq 0$, we conclude that a curve $X_t$ governed by Nesterov flow (4.1) has convergence guarantee:

$$f(X_t) - f^* \leq \frac{D_h(x^*, X_0)}{C t^p} = O \left( \frac{1}{t^p} \right)$$  \hfill (4.6)

which matches the $O(1/\epsilon k^p)$ convergence rate of $\mathcal{G}_p$ (3.11) in discrete time, as claimed. But the bound (4.6) holds for all $p > 0$, and only requires convexity of $f$ (in (4.5)) and $h$ (so that $D_h \geq 0$).

We note, (4.3) is a generalization of the energy functional in [13], who were the first to point out that accelerated gradient descent ($p = 2$, Euclidean case) in continuous time corresponds to the second order equation $\ddot{X}_t + \frac{3}{2} \dot{X}_t + \nabla f(X_t) = 0$, and proved a matching $O(1/t^2)$ convergence rate [13, Theorem 3.2]. They also remarked on the significance of 3 as being the smallest value of $r$ such that the modified equation $\ddot{X}_t + \frac{r}{2} \dot{X}_t + \nabla f(X_t) = 0$ has the same inverse quadratic $O((r-1)/t^2)$ convergence rate, and this guarantee breaks for $r < 3$, so there is a “phase transition” at $r = 3$.  

20
We can explain the “phase transition” as follows, setting \( r = p + 1 \). If we use \( \frac{t+1}{t} \dot{X}_t \) as the velocity term in (4.1), then we should increase the weight of \( \nabla f(X_t) \) to \( \frac{t^2}{t} t^{p-2} \) in order to get the optimal convergence rate \( O(1/t^p) \) (4.6). Indeed, we can generalize the energy functional (4.3) to 

\[ \mathcal{E}_t' = \rho_t(f(X_t) - f^*) + D_h(x^*, X_t + t_\mathcal{T} \dot{X}_t) \]

for any increasing \( \rho_t > 0 \) with \( \rho_0 = 0 \). By the same calculation (4.4), if \( \dot{X}_t + \frac{t+1}{t} \dot{X}_t + \frac{t^2}{t} \rho_t \nabla f(X_t) = 0 \), then \( \mathcal{E}_t' \leq 0 \) as long as \( \rho_t \leq C t^p \), yielding convergence rate \( O(1/\rho_t) \). In particular, the equation \( \dot{X}_t + \frac{t}{t} \dot{X}_t + \nabla f(X_t) = 0 \) from [13] is the case \( \rho_t = t^2/(r-1)^2 \) with convergence rate \( O(1/\rho_t) = O((r-1)^2/t^2) \), consistent with their result. So indeed 3 is special, as \( 3 = p + 1 \) when \( p = 2 \). Using \( r = p + 1 \geq 3 \) requires weighting \( \nabla f(X_t) \) by \( \frac{t^2}{t} t^{p-2} \), which is necessary for the \( O(1/t^2) \) rate in continuous time.

### 4.2 Discretizing Nesterov flow

In this section we examine how to discretize Nesterov flow (4.1) so as to preserve the convergence guarantee. Following the approach in Section 2.3, we choose to discretize the equivalent equation (4.2), which can be written as a system of two first order equations:

\[
\begin{align*}
Z_t &= X_t + \frac{t}{p} \dot{X}_t \quad \text{(4.7a)} \\
d_t \nabla h(Z_t) &= -C p t^{p-1} \nabla f(X_t). 
\end{align*}
\]

Now suppose we discretize \( X_t, Z_t \) into sequences \( x_k, z_k \) with time step \( \delta > 0 \), that is, if \( x_k = X_t \) then \( x_{k+1} = X_{t+\delta} = X_t + \delta \dot{X}_t \), and similarly for \( z_k = Z_t \), \( z_{k+1} = Z_{t+\delta} = Z_t + \delta \dot{Z}_t \). This means \( k \) discrete iterations, each corresponding to a jump of length \( \delta \), are equivalent to the elapse of \( t = \delta k \) continuous time.

Under this identification, (4.7a) becomes \( z_k = x_k + \frac{p+1}{p+\delta} (x_{k+1} - x_k) \), or equivalently:

\[
x_{k+1} = \frac{p}{k} z_k + \frac{k-p}{k} x_k \quad \text{(4.8)}
\]

which is the same as the \( x \)-update in \( \mathcal{G}_p \) (3.2c), except here we use \( x_k \) instead of \( y_k \) (which is currently not in the algorithm). Moreover, (4.8) uses convex weight \( \frac{p}{k} \), but it is equivalent to the weight \( \frac{p}{k+t} = \frac{p}{k} + o(\frac{1}{k}) \) in (3.2c). Note, in (4.8) there is no \( \delta \).

Similarly, (4.7b) becomes \( \frac{1}{\delta} (\nabla h(z_{k+1}) - \nabla h(z_k)) = -C p (\delta k)^{p-1} \nabla f(x_k) \), which we can recognize as the optimality condition of a mirror descent step:

\[
z_{k+1} = \arg\min_z \left\{ C p k^{p-1} \langle \nabla f(x_k), z \rangle + \frac{1}{\delta^p} D_h(z, z_k) \right\} \quad \text{(4.9)}
\]

which is the same as the \( z \)-update in \( \mathcal{G}_p \) (3.4), except here we use \( \nabla f(x_k) \) instead of \( \nabla f(y_k) \); moreover, (4.9) uses \( k^{p-1} \) instead of the equivalent weighting \( (k+1)^{(p-1)} = \Theta(k^{p-1}) \) in (3.4). We also see the scaling \( \epsilon = \delta^p \) of the step size \( \epsilon \) in (4.9) and the time step \( \delta \) in the discretization.

In principle, the two updates (4.8), (4.9) define an algorithm that “implements” Nesterov flow (4.7) in discrete time. However, we also want a matching convergence rate guarantee \( O(1/\epsilon k^p) \) (since \( \epsilon k^p = (\delta k)^p = t^p \)), and unfortunately that doesn’t seem possible with only (4.8), (4.9). We
can try to follow the approach in the proof of Theorem 6, and attempt to establish upper and lower estimates of the objective $f$ by the estimate function $\psi_k$. However, a key step in the proof is showing that the remainder of the expression (3.10) is nonnegative, for which we need the result (3.5) in Lemma 4 as well as the sequence $y_k$. Thus, we can view the accelerated gradient algorithm $\vec{G}^p$ (3.2) as a discretization of Nesterov flow (4.7), with the introduction of an additional sequence $y_k$ (3.2a) whose purpose is to guarantee inequality (3.5), which implies the matching convergence rate $O(1/\epsilon k^p)$.

It is curious that we need the sequence $y_k$ satisfying (3.5) to make the convergence proof work in discrete time. This $y_k$ differs from $x_k$ by a smaller time scale $\epsilon_{p-1}^p < \epsilon_1^p = \delta$ (3.18), so as $\delta \to 0$, $x_k$ and $y_k$ have the same continuous time limit $X_t = Y_t$. We also note that inequality (3.5) is the only place where the $(p-1)$-st order smoothness of $f$ is needed (by $\vec{G}^p$, which is used by $y_k$ (3.2a)). It would be interesting to see whether it is possible to replace $\vec{G}^p$ in (3.2a) by another algorithm that guarantees the same inequality (3.5) under a weaker assumption.

4.3 Interpretation as Euler-Lagrange equation

Nesterov flow (4.1) looks similar to the second order damped harmonic oscillator equation from physics, but with a subtle difference. Recall, in classical mechanics we typically model friction as a velocity-dependent force. The equation of motion is then a second order equation involving both time derivatives of $X_t$, e.g., $\ddot{X}_t + 2\zeta \dot{X}_t + X_t = 0$ for damped harmonic oscillator with “damping ratio” $\zeta > 0$; whereas the ideal (frictionless) harmonic oscillator $\ddot{X}_t = -X_t$ involves no velocity term. The presence of $2\zeta \dot{X}_t$ in damped harmonic oscillator changes the nature of the system, from conservative (conserves energy $E_t = \frac{1}{2}\|\dot{X}_t\|^2 + \frac{1}{2}\|X_t\|^2$, so ideal harmonic oscillator never stops) to dissipative (dissipates energy, the system stabilizes, and the curve $X_t$ converges).

Like damped harmonic oscillator, Nesterov flow (4.1) is also a second order equation involving both time derivatives of $X_t$, although note that the velocity term in Nesterov flow has time-varying coefficient $\frac{p+1}{p}^{-1} t^p$. Nevertheless, we can still capture Nesterov flow under the same general framework of dissipative system, but with “logarithmic damping” (instead of linear); see Section 5.

Concretely, we observe that Nesterov flow can be interpreted as the Euler-Lagrange equation:

$$\frac{d}{dt} \left\{ \frac{\partial}{\partial v} \mathcal{L}(X_t, \dot{X}_t, t) \right\} = \frac{\partial}{\partial x} \mathcal{L}(X_t, \dot{X}_t, t)$$

(4.10)

where $\mathcal{L}(x,v,t)$ is the (Nesterov) Lagrangian functional:

$$\mathcal{L}(x,v,t) = pt^{p-1} D_h \left( x + \frac{t}{p} v, x \right) - Cpt^{2p-1} f(x)$$

(4.11)

defined for any point $x \in \mathcal{X}$, tangent vector $v \in T_x \mathcal{X}$, and time $t \in \mathbb{R}$. In (4.10), $\frac{\partial}{\partial v} \mathcal{L}(X_t, \dot{X}_t, t)$ is the partial derivative of $\mathcal{L}(x,v,t)$ with respect to $v$ evaluated at $(x,v,t) = (X_t, \dot{X}_t, t)$, and similarly for $\frac{\partial}{\partial x} \mathcal{L}(X_t, \dot{X}_t, t)$. For the Nesterov Lagrangian (4.11), we can calculate these derivatives explicitly and verify that (4.10) above is indeed equivalent to Nesterov flow (4.1).
Recall, the Euler-Lagrange equation (4.10) is a necessary (and often sufficient) condition for $X_t$ to be a stationary point for the following variational problem, for any time $t_0 < t_1$ [4, Theorem 1]:

$$\min \int_{t_0}^{t_1} \mathcal{L}(X_t, \dot{X}_t, t) \, dt$$

where the minimization is performed over all continuously differentiable curves $X_t$ with fixed endpoints $X_{t_0} = x_0, X_{t_1} = x_1 \in \mathcal{X}$. The objective function in (4.12) is typically called the action $\mathcal{A}(X)$, and the problem (4.12) is known as the principle of least action, which has played an important role as an equivalent reformulation of much of classical physics.\(^9\)

Potential external connections aside, the significance of this observation is in making us aware of the rich structure in the family of Nesterov flows. Indeed, the interpretation of Nesterov flow as Euler-Lagrange equation allows us access to the techniques and results from (for example) calculus of variations, which may not be applicable to the first order rescaled gradient flows.\(^10\) And we believe that the discrete time accelerated algorithm $\vec{G}_p (3.2)$—whose convergence proof at first glance looks like an “algebraic trick”—is actually exploiting this structure.

Furthermore, it turns out that the family of Nesterov Lagrangians (4.11) is particularly nice, and in fact can be extended to a larger class of Lagrangians that preserves much of the nice properties.

## 5 A Lagrangian view of acceleration

We introduce the family of Bregman Lagrangians:

$$\mathcal{L}_{\alpha, \beta, \gamma}(x, v, t) = e^{\gamma t} \left( e^{2\alpha t} D_h \left( x + e^{-\alpha t} v, x \right) - e^{\beta t} f(x) \right)$$

where $\alpha_t \in \mathbb{R}$ is the scale function, $\beta_t \in \mathbb{R}$ the weight function, and $\gamma_t \in \mathbb{R}$ the damping function. In the Euclidean case $h(x) = \frac{1}{2} \|x\|^2 (\ell_2$-norm), the Lagrangian (5.1) simplifies to (notice no $\alpha_t$):

$$\mathcal{L}_{\alpha, \beta, \gamma}(x, v, t) = e^{\gamma t} \left( \frac{1}{2} \|v\|^2 - e^{\beta t} f(x) \right).$$

### Ideal scaling.

In general, $\alpha_t, \beta_t, \gamma_t$ in (5.1) can be arbitrary, but we find there is an ideal scaling that is necessary for some results to hold (e.g., (5.3b) simplifies (5.4) to (5.5)):

$$\hat{\beta}_t = 2\dot{\alpha}_t + e^{\alpha t}$$

$$\hat{\gamma}_t = -\dot{\alpha}_t + e^{\alpha t}.\quad (5.3b)$$

\(^9\)Chiefly among them Newton’s law of motion $\ddot{X}_t = -\nabla f(X_t)$, which comes from the ideal Lagrangian $\mathcal{L}_0(x, v, t) = \frac{1}{2} \|v\|^2 - f(x)$, where here $f$ is the “potential” function generating the “force” $F(x) = -\nabla f(x)$.

\(^{10}\)Although, we can interpret rescaled gradient flow as the “massless limit” of a Lagrangian flow (Section 5.2).
**Euler-Lagrange equation.** For general functions $\alpha_t, \beta_t, \gamma_t$, the Euler-Lagrange equation (4.10)\[ \frac{\partial}{\partial t} \{ \frac{\partial}{\partial \dot{X}_t} \mathcal{L}(X_t, \dot{X}_t, t) \} = \frac{\partial}{\partial X_t} \mathcal{L}(X_t, \dot{X}_t, t) \] for the Bregman Lagrangian $\mathcal{L} \equiv \mathcal{L}_{\alpha, \beta, \gamma}$ (5.1) is given by:

\[ \ddot{X}_t + (-\dot{\alpha}_t + e^{\alpha t}) \dot{X}_t - e^{\beta t} \nabla^2 h \left( X_t + e^{-\alpha t} \dot{X}_t \right)^{-1} \nabla f(X_t) + e^{\alpha t} (\dot{\gamma}_t - \dot{\alpha}_t + e^{\alpha t} \dot{X}_t) \nabla^2 h \left( X_t + e^{-\alpha t} \dot{X}_t \right)^{-1} \left( \nabla h \left( X_t + e^{-\alpha t} \dot{X}_t \right) - \nabla h(X_t) \right) = 0. \] (5.4)

If $\gamma_t$ satisfies the ideal scaling (5.3b), then the Euler-Lagrange equation (5.4) simplifies to:

\[ \ddot{X}_t + \dot{\gamma}_t \dot{X}_t + e^{\beta t} \nabla^2 h \left( X_t + e^{-\alpha t} \dot{X}_t \right)^{-1} \nabla f(X_t) = 0 \] (5.5)

which we call Bregman flow. In the Euclidean case (5.2), it simplifies to $\ddot{X}_t + \dot{\gamma}_t \dot{X}_t + e^{\beta t} \nabla f(X_t) = 0$.

**Convergence rate via energy functional.** Generalizing (4.3), we define the energy functional:

\[ \mathcal{E}_t = e^{\beta t - 2\alpha t} (f(X_t) - f^*) + D_h \left( x^*, X_t + e^{-\alpha t} \dot{X}_t \right). \] (5.6)

Assuming the ideal scaling $\dot{\gamma}_t = -\dot{\alpha}_t + e^{\alpha t}$ (5.3), $\mathcal{E}_t$ has time derivative:

\[ \dot{\mathcal{E}}_t = (\dot{\beta}_t - 2\dot{\alpha}_t) e^{\beta t - 2\alpha t} (f(X_t) - f^*) + e^{\beta t - 2\alpha t} \nabla f(X_t) \dot{X}_t + e^{-\alpha t} \left( \nabla^2 h \left( X_t + e^{-\alpha t} \dot{X}_t \right) \dot{X}_t + \dot{\gamma}_t \dot{X}_t, X_t - x^* + e^{-\alpha t} \dot{X}_t \right). \]

If $X_t$ satisfies the Euler-Lagrange equation (5.5), then $\dot{\mathcal{E}}_t$ simplifies to:

\[ \dot{\mathcal{E}}_t = (\dot{\beta}_t - 2\dot{\alpha}_t) e^{\beta t - 2\alpha t} (f(X_t) - f^*) - e^{\beta t - \alpha t} \nabla f(X_t) (X_t - x^*). \]

Now invoking the convexity of $f$ (1.12), we can bound the expression above by:

\[ \dot{\mathcal{E}}_t \leq e^{\beta t - 2\alpha t} \left( \dot{\beta}_t - 2\dot{\alpha}_t - e^{\alpha t} \right) (f(X_t) - f^*). \] (5.7)

Thus, we see that if $\dot{\beta}_t \leq 2\dot{\alpha}_t + e^{\alpha t}$, then $\dot{\mathcal{E}}_t \leq 0$, which implies $\mathcal{E}_t \leq \mathcal{E}_0$ for all $t \geq 0$. Since $h$ is convex, $D_h(x^*, X_t + e^{-\alpha t} \dot{X}_t) \geq 0$, therefore we get the bound $e^{\beta t - 2\alpha t} (f(X_t) - f^*) \leq \mathcal{E}_0$. This gives a convergence rate $\rho_t = 2\alpha_t - \beta_t$, which we summarize in the following theorem. Note, for any $\alpha_t$, the optimal choice of $\beta_t$ in (5.8) below is given by the ideal scaling (5.3a), which yields rate $\rho_t = 2\alpha_t - \beta_t = \int_0^t e^{\alpha s} ds$.

**Theorem 7.** If $f$ and $h$ are convex and the ideal scaling (5.3b) holds, then for any $\alpha_t, \beta_t$ satisfying $\dot{\beta}_t \leq 2\dot{\alpha}_t + e^{\alpha t}$, the curve $X_t$ governed by the Euler-Lagrange equation (5.5) has convergence rate $\rho_t = 2\alpha_t - \beta_t$:

\[ f(X_t) - f^* \leq \frac{\mathcal{E}_0}{e^{\beta t - 2\alpha t}} = O \left( e^{2\alpha t - \beta t} \right). \] (5.8)
Example: Nesterov Lagrangian. As our motivating example, Nesterov Lagrangian (4.11) is a special case of the Bregman Lagrangian (5.1) when we choose:

\[
\begin{align*}
\alpha_t &= -\log t + \log p \quad (5.9a) \\
\beta_t &= (p - 2) \log t + 2 \log p + \log C \quad (5.9b) \\
\gamma_t &= (p + 1) \log t - \log p \quad (5.9c) \\
\rho_t &= p \log t \quad (5.9d)
\end{align*}
\]

which satisfy the ideal scaling (5.3)\(^{11}\) for any \(p > 0\). With these parameter choices, we can verify that Bregman Lagrangian (5.1) reduces to Nesterov Lagrangian (4.11), Bregman flow (5.5) reduces to Nesterov flow (4.1), and the convergence rate (5.8) recovers our earlier result (4.6).

However, the Bregman Lagrangian family (5.1) is much more general, and we wish to study its more general properties (which we can then specialize to any subfamily, including Nesterov (4.11)). To expand the repertoire of Bregman Lagrangians, in Section 6 we study the family of constant \(\alpha_t = \log c\) with rate \(\rho_t = ct\), and show connections to the restart scheme proposed by Nesterov [9] to obtain linear convergence in discrete time under uniform convexity assumption.

In the rest of this section, we discuss the interpretation of Bregman Lagrangian as approximating the “true” momentum method (Hessian Lagrangian); we will also see how to interpret rescaled gradient flow as the massless limit of a (modified) Lagrangian flow.

5.1 Bregman Lagrangian as an approximation of Hessian Lagrangian

We define the family of Hessian Lagrangian:

\[
\mathcal{L}_{\beta,\gamma}(x, v, t) = e^\gamma \left( \frac{1}{2} \|v\|^2_{h(x)} - e^\beta f(x) \right) \quad (5.10)
\]

where as before \(\beta_t \in \mathbb{R}\) is the weight function and \(\gamma_t \in \mathbb{R}\) the damping function. The Hessian Lagrangian (5.10) is a damped and weighted version of the ideal Lagrangian (with Hessian metric) \(\mathcal{L}_0(x, v, t) = \frac{1}{2} \|v\|^2_{h(x)} - f(x)\).

The Bregman divergence, being a first order approximation error, can be seen as approximating the squared Hessian norm:

\[
e^{2\alpha} D_h \left( x + e^{-\alpha} v, x \right) \approx \frac{1}{2} \|v\|^2_{h(x)} \quad (5.11)
\]

for any \(x \in \mathcal{X}, v \in T_x\mathcal{X}\), and \(\alpha \in \mathbb{R}\) (such that \(x + e^{-\alpha} v \in \mathcal{X}\)). Therefore, we can interpret Bregman Lagrangian (5.1) as approximating the Hessian Lagrangian (5.10): \(\mathcal{L}_{\alpha,\beta,\gamma} \approx \mathcal{L}_{\beta,\gamma}\), for all \(\alpha_t, \beta_t, \gamma_t\). Note in the Euclidean case (5.11) is an equality so Bregman and Hessian Lagrangians coincide.

The Euler-Lagrange equation for the Hessian Lagrangian (5.10) is given by:

\[
\frac{1}{2} \nabla^2 h(X_t) \ddot{X}_t + \nabla h(X_t) \left( \dot{X}_t + \gamma_t \dot{X}_t \right) + e^\beta \nabla f(X_t) = 0 \quad (5.12)
\]

\(^{11}\)Note, (5.3) only determines \(\beta_t, \gamma_t\) up to constant terms, but we will see in Section 7.2 that (5.9) is the proper choice of constants from the perspective of time dilation.
where the third order derivative $\nabla^3 h$ comes from being the derivative of the metric tensor $\nabla^2 h$. Notice if we remove the first term from (5.12), then we recover the Euler-Lagrange equation (5.5) for the Bregman Lagrangian in the case $\alpha_t = \infty$ (which is the ideal case since by L'Hôpital's rule, $\lim_{\alpha \to \infty} e^{2\alpha} D_h(x + e^{-\alpha} v, x) = \frac{1}{2} ||v||^2_{h(x)}$). Thus, Bregman flow (5.5) can be interpreted as an approximation to the Hessian flow (5.12) that removes the $\nabla^3 h$ term (and compensates by using $\nabla^2 h(X_t + e^{-\alpha_t} \dot{X}_t)$). Note, Bregman flow can be equivalently written as (4.7), which can then be discretized using mirror descent to yield an algorithm (3.2), which does not require $\nabla^2 h$ but only $\nabla h$. Therefore, this offers an interpretation of Nesterov's acceleration technique as a clever approximate discretization of the Hessian Lagrangian, which reduces the complexity of the required computation from $\nabla^3 h$ in (5.12) down to $\nabla h$ in (3.2).

However, it is actually still unclear why the Hessian Lagrangian is the right thing to approximate. For example, we do not have any convergence guarantee on the Hessian flow. The convergence rate $\rho_t = \int_0^t e^{\alpha_t} ds$ (5.8) for Bregman Lagrangian tends to $\infty$ as $\alpha \to \infty$. This suggests that in the ideal limit $\alpha_t = \infty$, Hessian flow has instantaneous convergence $(f(X_t) - f^* \leq 0$ for any $t > 0)$, although note also that under the ideal scaling (5.3), $\beta, \gamma \to \infty$ as $\alpha \to \infty$ so this limit is not well defined.

Let us take a step back and notice, in approximating Hessian norm by Bregman divergence (5.11) we have introduced a scale variable $\alpha \in \mathbb{R}$, which provides the conversion factor between the scales of the point $x \in X$ and the tangent vector $v \in T_x X$. Ordinarily, we treat $v$ as operating at a small (infinitesimal) scale $\varepsilon > 0$ where linear approximation holds, e.g., $f(x + \varepsilon v) = f(x) + \varepsilon (\nabla f(x), v)$. But in practice, how should we choose $\alpha$? As noted, the ideal is $\alpha = \infty$, in which case Bregman Lagrangian reduces to Hessian Lagrangian. However, as soon as $\alpha = \log(1/\varepsilon) < \infty$, there is the ideal scaling (5.3) in Bregman Lagrangian that binds $\alpha, \beta, \gamma$ together, and renders the limit $\alpha \to \infty$ nonsensical. But in return, the ideal scaling gives us convergence rate $\rho_t = \int_0^t e^{\alpha_t} ds$, which is better for larger $\alpha_t$. For example, Nesterov Lagrangian (4.11) uses logarithmic $\alpha_t = -\log t + \log p$, which yields $p$-sublinear rate $\rho_t = p \log t$. The exponential analog of Nesterov in Section 6.1 uses constant $\alpha_t = \log c$, which yields linear rate $\rho_t = ct$.

### 5.2 Rescaled gradient flow as massless limit of Lagrangian flow

We define the $p$-th power Lagrangian, $p > 0$:

$$\mathcal{L}(x, v, t) = e^{t/m} \left( \frac{m}{p} ||v||^p - f(x) \right)$$

(5.13)

where $m > 0$ is the mass of our (fictitious) particle. In Section 5 we implicitly set $m = 1$, but here we are interested in the limiting behavior $m \to 0$, with $p$ fixed.

The Euler-Lagrange equation $\frac{d}{dt} \left( \frac{\partial}{\partial X_t} \mathcal{L}(X_t, \dot{X}_t, t) \right) = \frac{\partial}{\partial X_t} \mathcal{L}(X_t, \dot{X}_t, t)$ for the Lagrangian (5.13) is:

$$||\dot{X}_t||^{p-2} (m \ddot{X}_t + \dot{X}_t) + m(p-2)\|\dot{X}_t\|^{p-4} \langle \ddot{X}_t, \dot{X}_t \rangle \dot{X}_t + \nabla f(X_t) = 0.$$  (5.14)

So as $m \to 0$, the Euler-Lagrange equation (5.14) converges to the rescaled gradient flow (2.12):

$$||\dot{X}_t||^{p-2} \dot{X}_t + \nabla f(X_t) = 0.$$
This gives an interpretation of rescaled gradient flow—which is a first order equation—as the massless limit of the Lagrangian flow (5.14), which is a second order equation. However, notice that this massless limit also corresponds to infinite momentum: $e^{t/m}m||\dot{X}_t||^{p-2}\dot{X}_t \to \infty$ as $m \to 0$. This means as $m \to 0$, our particle (whose evolution is governed by (5.14)) becomes infinitely massive. In the limit $m = 0$ (rescaled gradient flow) there is no oscillation; the particle just rolls downhill (with infinite “friction”) and stops at the minimum $x^*$ as soon as the force $-\nabla f$ vanishes.

Note, this may run contrary to the idea that adding an acceleration/momentum term amounts to preventing oscillation, hence the faster convergence. What our interpretation suggests is the opposite: The first order rescaled gradient flow is the case of infinite momentum and no oscillation (rather than no momentum and big oscillation), and the effect of moving to the second order Lagrangian flow is to unwind the curve (to finite momentum) where it travels faster, but there is now oscillation. For example, the case $p = 2$ in (5.14) is the damped harmonic oscillator (when $f(x) = \frac{1}{2}||x||^2$). This point of view is also consistent with the work [11], who addressed this oscillation issue by a restart scheme (cf. Section 6.2). See also Appendix A.4 for an interpretation of natural gradient flow as the massless limit of Hessian Lagrangian flow.

6 Linear convergence rate via uniform convexity

We study the exponential analog of Nesterov Lagrangian, which uses constant scale function and has linear convergence rate. We also show how to extend Nesterov’s restart scheme [9] in discrete time to get linear convergence rate when the objective function is uniformly convex.

6.1 Exponential Nesterov Lagrangian family

We define the exp-Nesterov Lagrangian, for any $c > 0$:

$$\mathcal{L}(x, v, t) = ce^{ct} \left( D_h \left( x + \frac{1}{c}v, x \right) - e^{ct}f(x) \right).$$

This is the Bregman Lagrangian (5.1) with the following choices, which satisfy the ideal scaling (5.3):

\begin{align*}
\alpha_t &= \log c \\
\beta_t &= ct + 2 \log c \\
\gamma_t &= ct - \log c \\
\rho_t &= ct.
\end{align*}

The Euler-Lagrange equation for the Lagrangian (6.1) is given by:

$$\ddot{X}_t + c\dot{X}_t + c^2 e^{ct} \nabla^2 h \left( X_t + \frac{1}{c}\dot{X}_t \right)^{-1} \nabla f(X_t) = 0.$$  

By Theorem 7, the exp-Nesterov flow $X_t$ (6.3) has linear convergence rate $\rho_t = ct$ for convex $f$:

$$f(X_t) - f^* \leq \frac{f(X_0) - f^* + D_h(x^*, X_0 + \frac{1}{c}\dot{X}_t)}{e^{ct}} = O(e^{-ct}).$$
Thus, whereas Nesterov flow (4.1) has sublinear rate, exp-Nesterov flow (6.3) has linear rate.

Let us examine how to discretize exp-Nesterov flow. As in Section 4.2, we first write (6.3) as:

\[ Z_t = X_t + \frac{1}{c} \dot{X}_t \]  
\[ \frac{d}{dt} \nabla h(Z_t) = -ce^{ct} \nabla f(X_t). \]

We discretize \( X_t, Z_t \) into sequences \( x_k, z_k \) with time step \( \delta > 0 \) as before, so \( t = \delta k \). Using mirror descent to implement (6.5b), we obtain discrete time equations similar to (3.2c), (3.4):

\[ z_k = \arg \min_z \left\{ ce^{\delta k} \langle \nabla f(x_k), z \rangle + \frac{1}{\delta} D_h(z, z_{k-1}) \right\} \]  
\[ x_{k+1} = c\delta z_k + (1 - c\delta)x_k. \]

Note, the weight in (6.6b) is independent of time, but depends on \( \delta \); and (6.6a) suggests the step size \( \epsilon = \delta \) in the algorithm. If our analogy between continuous and discrete time convergence holds, then given the \( O(e^{-ct}) \) convergence rate (6.4) for \( X_t \), we expect a matching \( O(\frac{1}{\epsilon} e^{-c\delta}) \) convergence rate in discrete time.

However, it is not clear how to get that with only (6.6). If we try to adapt the proof of Theorem 6 (with the ideal choice \( p = \infty \)), then we find that we need to introduce a sequence \( y_k \) satisfying the following analog of Lemma 4, in order to conclude a convergence rate \( O(\delta e^{-c\delta}) \):

\[ \langle f(y_k), x_k - y_k \rangle \geq \delta e^{-c\delta} \| \nabla f(y_k) \|_* \]  

Notice, the rates above are consistent if we set \( \epsilon = \delta = 1 \). But the condition (6.7) means we need to make a constant improvement in each iteration from \( x_k \) to \( y_k \), although we are also free to be creative with how to construct \( y_k \) and impose any assumption on \( f \).

### 6.2 Restart scheme for uniformly convex objective function

We present a generalization of Nesterov’s restart scheme [9] that obtains a linear convergence rate when the objective function is both smooth and uniformly convex. This in a sense can be seen as a discrete time version of exp-Nesterov flow (6.5), which implements the exponential weighting and the improvement requirement (6.7) by running the accelerated gradient algorithm \( \hat{G}_p \) (3.2) for some amount of time, within each iteration.

**Linear convergence rate for \( G_p \).** Following [9, Section 5], we first show that the \( p \)-th gradient method \( G_p \) has linear convergence rate if \( f \) is uniformly convex. Concretely, consider the following \( G_p \) with larger regularization (like (3.2a)), for \( p \geq 2 \):

\[ x_{k+1} = \arg \min_x \left\{ f_{p-1}(x; x_k) + \frac{1}{p} \epsilon \| x - x_k \|_p^p \right\} \]  

where \( \epsilon > 0 \) here is a fixed step size. By Lemma 4, we know that if \( f \) is \( \frac{(p-1)!}{\epsilon} \)-smooth of order \( p-1 \), then \( \langle \nabla f(x_k+1), x_k - x_{k+1} \rangle \geq \frac{1}{4} \epsilon \frac{1}{p-1} \| \nabla f(x_{k+1}) \|_1^{p-1} \). If \( f \) is \( \sigma \)-uniformly convex of order \( p \), then we
also have \([9, \text{Lemma 3}]\) \(\|\nabla f(x_{k+1})\|^p \geq \frac{p}{p-1} \sigma^{1/(p-1)} (f(x_{k+1}) - f^*)\). Moreover, if \(f\) is convex \((1.12)\), then:

\[
f(x_{k+1}) - f^* \leq \frac{f(x_k) - f^*}{1 + \frac{1}{4}(\epsilon \sigma)^{1/(p-1)}} \leq \frac{f(x_1) - f^*}{(1 + \frac{1}{4}(\epsilon \sigma)^{1/(p-1)})^k}.
\]

If the \((\text{inverse})\) condition number \(\kappa = \epsilon \sigma\) is small, then \(1 + \frac{1}{4}(\epsilon \sigma)^{1/(p-1)} \approx e^{\kappa^{1/p-1}/4}\). And again by the smoothness of \(f\), we have \(f(x_1) \leq \min_x \{f_{p-1}(x; x_0) + \frac{2}{c_p} \|x - x_0\|^p\} \leq f^* + \frac{3}{c_p} \|x_0 - x^*\|^p\).

Therefore, \((6.9)\) yields the convergence rate \(\rho_k = ck\), \(c = \frac{1}{4} \kappa^{1/p}\), for the \(p\)-th gradient method \(G_p\) \((6.8)\), generalizing the result of \([9, (5.6)]\):

\[
f(x_{k+1}) - f^* \leq \frac{3 \|x_0 - x^*\|^p}{ep(1 + \frac{1}{4} \kappa^{1/p})^k} = O\left(\frac{1}{e} e^{-\kappa^{1/p} k/4}\right)
\]

which matches the desired convergence rate \(O(\frac{1}{e} e^{-ck})\) discussed in Section 6.1.

**Improved linear convergence rate for \(G_p\) with restart.** We now show that a variant of the accelerated gradient method \(G_p\) attains a better linear convergence rate than \(G_p\) \((6.8)\). Specifically, consider the following restart scheme, generalizing \([9, (5.7)]\):

\[
x_{(k+1)m} = (\text{the output } \hat{y}_m\text{ of running } G_p\ (3.2) \text{ for } m \text{ iterations with input } \hat{x}_0 = x_{km})
\]

where \(m = 24p/\kappa^{1/p}\), and \(\kappa = \epsilon \sigma\) as before is the inverse condition number of \(f\). Here we assume we replace the Bregman divergence in the \(z\)-update \((3.2b)\) by \(d_p(z) = \frac{1}{p} \|z - x_0\|^p\), rescaled by its uniform convexity constant \(2^{-p+2}\) \((3.1)\). The proof of Theorem 6 still holds in this case, and for concreteness we choose \(C = (4p)^{-p}\).

Then, since \(f\) is \(\sigma\)-uniformly convex of order \(p\) \((1.15)\), and by the bound \((3.11)\) from Theorem 6:

\[
\frac{\sigma}{p} \|x_{(k+1)m} - x^*\|^p \leq f(x_{(k+1)m}) - f^* \leq \frac{(4p)^p 2^{p-2} \|x_{km} - x^*\|^p}{\epsilon m^{(p)}} \leq \frac{\sigma}{p e} \|x_{km} - x^*\|^p
\]

where the last inequality follows from our choice of \(m\). Iterating \((6.12)\) and rescaling the index \(k \equiv \frac{k}{m}\), we obtain \(\|x_k - x^*\|^p \leq e^{-k/m} \|x_0 - x^*\|^p\). To convert this into a bound on the function value, we use the smoothness of \(f\). Let \(y_k\) be the output of \(G_p\) \((6.8)\) with input \(x_k\). As noted before, if \(f\) is \((p-1)!\)-smooth of order \(p - 1\), then \(f(y_k) - f^* \leq \frac{3}{c_p} \|x_k - x^*\|^p\). Therefore, we conclude that:

\[
f(y_k) - f^* \leq \frac{3 \|x_0 - x^*\|^p}{\epsilon p e^{k/m}} = O\left(\frac{1}{e} e^{-\kappa^{1/p} k/4p}\right)
\]

which matches the convergence rate \(O(\frac{1}{e} e^{-ck})\) as discussed in Section 6.1 with \(c = \frac{1}{24p} \kappa^{1/p}\). Note, this linear rate \(\rho_t = c_k \kappa^{1/p}\) has better dependence for small \(\kappa = \epsilon \sigma\) than \((6.10)\), generalizing the conclusion of \([9, (5.8)]\). However, the link to continuous time is not as clear as that of the Nesterov family.

29
7 Time dilation: Faster convergence by speeding up time

In this section we introduce the idea of time dilation, and show that a large family of Bregman Lagrangians (which include Nesterov and exp-Nesterov) can be interpreted as the result of speeding up any single curve.

7.1 Time dilation

We recall the argument in Section 1 that optimization in continuous time is easy because we can get arbitrarily fast convergence. The idea is that once we have a curve $X_t$ that converges at some rate $\rho(t)$, then we can speed it up to $Y_t = X_{\tau(t)}$ with improved convergence rate $\rho(\tau(t)) > \tau(t)$.

Here $\tau = \tau(t) \in \mathbb{R}$ is a time dilation: A smooth, strictly increasing (hence invertible) function defined on the time domain $\mathbb{R}$ (or a subset of it), whose inverse $\tau^{-1}$ is also smooth. The set $\mathcal{T}$ of time dilations forms a group under function composition. The identity element is the identity time:

$$t_{id}(t) = t \quad \forall t \in \mathbb{R} \quad (7.1)$$

which is the default time dilation we use, so normally time flows at unit speed: $dt_{id}/dt = 1$.

Traversing a curve $X_t$ at another time speed $Y_t = X_{\tau(t)}$ is equivalent to replacing the default time dilation $t_{id}$ by $\tau \in \mathcal{T}$, so now time flows at speed $d\tau/dt = \dot{\tau}(t)$. We say that $\tau$ is faster than $t_{id}$ if $\dot{\tau}(t) \geq 1$, in which case shifting from $t_{id}$ to $\tau$ amounts to speeding up time, and the convergence rate $\rho(\tau(t))$ of $Y_t$ is larger than the original rate $\rho(t)$ of $X_t$ (if $\rho(t)$ is increasing).\(^{12}\)

However, how meaningful is this idea? Recall, our main interest is in understanding the parallel behavior between continuous and discrete time optimization, so even if we have a fast rate in continuous time, it may not be of interest to us if we don’t know how to implement it in discrete time (with matching convergence rate).

Recall also from Section 1 the example of the gradient flow $\dot{X}_t = -\nabla f(X_t)$, which has convergence rate $\rho_t = -\log t$. The sped-up version $\dot{Y}_t = X_{\tau(t)}$ satisfies $\dot{Y}_t = -\dot{\tau}(t)\nabla f(Y_t)$, but is is not a gradient flow anymore (not of the form $\dot{Y}_t = -\nabla \tilde{f}(Y_t)$ for some function $\tilde{f}$ that does not explicitly depend on time). Similarly, speeding up rescaled gradient flow (2.10) yields a curve that is not in the rescaled gradient flow family. While these properties inhibit our understanding of how these curves relate to each other in continuous time, it turns out Bregman Lagrangian flows (5.5) have nice properties under time dilation, which we explore next.

7.2 Bregman Lagrangian family under time dilation

We show that Bregman Lagrangian family is closed under the action of the time dilation group $\mathcal{T}$. Note, in general it holds that speeding up a Lagrangian curve (i.e., Euler-Lagrange curve $X_t$ for a Lagrangian $\mathcal{L}$) results in another Lagrangian curve, so the space of general Lagrangian curves is closed under time dilation. Moreover, the Bregman Lagrangians (5.1) form a special subfamily of the Lagrangian space, and we can characterize precisely how they transform under time dilation.

\(^{12}\)But for convenience, we refer to $Y_t = X_{\tau(t)}$ as the sped-up version of $X_t$ regardless of whether $\tau$ is faster than $t_{id}$. 

30
We begin by noting that the sped-up curve $Y_t = X_{\tau(t)}$ has time derivatives:

\[
\begin{align*}
\dot{Y}_t &= \dot{\tau}(t) \dot{X}_{\tau(t)} \\
\ddot{Y}_t &= \ddot{\tau}(t) \dot{X}_{\tau(t)} + (\dot{\tau}(t))^2 \ddot{X}_{\tau(t)}.
\end{align*}
\]  

(7.2a) (7.2b)

Thus, if $X_t$ satisfies the Euler-Lagrange equation (5.5) for a Bregman Lagrangian $\mathcal{L} = \mathcal{L}_{\alpha,\beta,\gamma}$ (5.1), then $Y_t = X_{\tau(t)}$ also satisfies the Euler-Lagrange equation for the modified Lagrangian $\mathcal{L}^{(\tau)} = \mathcal{L}_{\alpha^{(\tau)}, \beta^{(\tau)}, \gamma^{(\tau)}}$, where the parameters $\alpha, \beta, \gamma$ are transformed to $\alpha^{(\tau)}, \beta^{(\tau)}, \gamma^{(\tau)}$:

\[
\begin{align*}
\alpha^{(\tau)}_t &= \alpha_{\tau(t)} + \log \dot{\tau}(t) \\
\beta^{(\tau)}_t &= \beta_{\tau(t)} + 2 \log \dot{\tau}(t) \\
\gamma^{(\tau)}_t &= \gamma_{\tau(t)} - \log \dot{\tau}(t).
\end{align*}
\]  

(7.3a) (7.3b) (7.3c)

This means each $\tau \in \mathcal{T}$ induces a map $\mathcal{L} \mapsto \mathcal{L}^{(\tau)}$ on the space of general Bregman Lagrangians:

\[
\mathcal{L} = \{ \mathcal{L}_{\alpha,\beta,\gamma} : \alpha, \beta, \gamma_{\tau} \in \mathbb{R} \}.
\]  

(7.4)

Furthermore, by chain rule, we see that the transformation (7.3) satisfies the composition property $(\alpha^{(\tau)}(\theta)) = \alpha^{(\tau \circ \theta)}$ for all $\tau, \theta \in \mathcal{T}$, and similarly for $\beta, \gamma$ (that is, speeding up by $\tau$ and then $\theta$ is equivalent to speeding up once by $\tau \circ \theta$). This lifts to the Lagrangian level: $(\mathcal{L}^{(\tau)})(\theta) = \mathcal{L}^{(\tau \circ \theta)}$. Formally, this means the mapping $\mathcal{L} \mapsto \mathcal{L}^{(\tau)}$ is a (right) group action of $\mathcal{T}$ on $\mathcal{L}$, namely, a group homomorphism from $\mathcal{T}$ to the permutation group of $\mathcal{L}$.

This conclusion extends to Hessian Lagrangian, which is the $\alpha \to \infty$ limit of Bregman Lagrangian; thus, $\mathcal{T}$ also acts on the space of Hessian Lagrangians with the same transformation rules (7.3b), (7.3c).\(^{13}\)

### 7.3 The orbit of ideal Bregman Lagrangians

The ideal scaling (5.3) defines a special “one-dimensional” subspace of ideal Bregman Lagrangians:

\[
\mathcal{L}_0 = \left\{ \mathcal{L}_{\alpha,\beta,\gamma} : \alpha \in \mathcal{A}, \beta = 2\alpha + \int e^\alpha, \gamma = -\alpha + \int e^\alpha \right\} \subseteq \mathcal{L}
\]  

(7.5)

where $\mathcal{A}$ is the space of all scale functions $\alpha_t \in \mathbb{R}$, and we use the shorthand $\int e^\alpha \equiv \int_0^t e^{\alpha_s} ds$.\(^{14}\) Recall by Theorem 7, the Lagrangian curve $X_t$ of an ideal Lagrangian $\mathcal{L}_\alpha \equiv \mathcal{L}_{\alpha,\beta,\gamma} \in \mathcal{L}_0$ has convergence rate $\rho_t = \int e^\alpha$.

We observe that the family of ideal Bregman Lagrangians is closed under the action of time dilation. Indeed, the ideal scaling (5.3) holds for $\alpha, \beta, \gamma$ if and only if it holds for $\alpha^{(\tau)}, \beta^{(\tau)}, \gamma^{(\tau)}$. Equivalently, $\mathcal{L}_\alpha \in \mathcal{L}_0$ if and only if $\mathcal{L}^{(\tau)} \in \mathcal{L}_0$ for any $\tau \in \mathcal{T}$.

\(^{13}\)Note, we can also show that $\mathcal{T}$ acts on the family of $p$-th power Lagrangian $\mathcal{L}_{\beta,\gamma}(x, v, t) = e^{\gamma t} (\frac{1}{p} ||v||^p - e^{\beta t} f(x))$ ((5.13) is the case $\gamma_t = t/m$ and $\beta_t = 0$), where now $\beta$ transforms as $\beta^{(\tau)}_t = \beta_{\tau(t)} + p \log \dot{\tau}(t)$ (7.3b) is case $p = 2$).

\(^{14}\)More generally, we can consider the halfplane defined by $\beta \leq 2\alpha + \int e^\alpha$, for which Theorem 7 holds.
Conversely, if we start from any ideal Lagrangian $L \in L_0$—say the standard Nesterov Lagrangian $L^* (p = 2$ in (4.11))—then we can reach any other ideal Lagrangian $L_\alpha$, $\alpha \in \mathcal{A}$, by choosing the time dilation function $\tau = e^{\frac{1}{2} \int e^\alpha}$.

Therefore, we conclude that the family of ideal Bregman Lagrangians (7.6) is an orbit under the action of the time dilation group $T$. That is, we can interpret $L_0$ as the result of speeding up any initial Lagrangian (for concreteness $L^*$), over all possible time dilations:

$$L_0 = \big\{ (L^*)^{(\tau)} : \tau \in T \big\}. \quad (7.6)$$

Notice, the convergence rate transforms consistently: $L^*$ has rate $\rho_t = 2 \log t$, so when we speed it up (by $\tau = e^{\frac{1}{2} \int e^\alpha}$) to $L_\alpha$, the rate transforms to $\rho_\tau = 2 \log \tau = \int e^\alpha$, as expected. Equivalently, we can also note that as $\alpha \mapsto \alpha^{(\tau)} = (\alpha \circ \tau) + \log \dot{\tau}$, the rate $\int e^\alpha = \int e^{\alpha \circ \tau} dt$ transforms to $\int \tau e^{(\alpha \circ \tau)} dt = \int e^\alpha d\tau$, consistent with the idea that we simply replace time $t (= t_{id})$ by $\tau$.

### 7.4 The orbit of Nesterov Lagrangians

We define the subgroup of polynomial time dilations $T_{pol} \subseteq T$:

$$T_{pol} = \{ \tau_p(t) = t^p : p > 0 \} \quad (7.7)$$

which is isomorphic to the multiplicative group $\mathbb{R}_{>0}$. The subgroup $T_{pol}$ inherits the action on $L_0$, but the action now partitions $L_0$ into (sub)orbits.

We observe, the family of Nesterov Lagrangians (4.11) forms an orbit under the action of $T_{pol}$. Indeed by (7.3a), if we start from $\alpha^* = - \log t + \log 2$ (for $L^*$ (4.11)), then for any $p > 0$, the time dilation $\tau(t) = t^{\frac{p}{2}}$ sends us to $\alpha^{*(\tau)} = - \log t + \log p$. Therefore, we can view the Nesterov flows (4.1) as the result of speeding up the Lagrangian flow of $L^* (p = 2)$, or any starting curve.

Moreover, as we have seen in Section 3, this speedup can be implemented in discrete time with matching convergence rate as the accelerated gradient methods $\tilde{G}_p$ (3.2). Thus, we can interpret the family of accelerated gradient methods $\tilde{G}$ as being the result of “speeding up” the algorithm $\tilde{G}_2$ (accelerated gradient descent) in discrete time—which we achieve via passage to continuous time, and at the cost of higher-order smoothness assumption on $f$.

### 7.5 The orbit of exp-Nesterov Lagrangians, isomorphic to Nesterov Lagrangians

From the standard Nesterov Lagrangian $L^* (p = 2$ in (4.11)), we can use time dilation $\tau = e^{\frac{1}{2} t}$ to reach the exp-Nesterov Lagrangian $L = L_c$ (6.1) with $\alpha = \log c$, which has linear rate $\rho_t = ct$, $c > 0$. Recall, from $L^*$ we can generate the Nesterov Lagrangians (4.11) as an orbit of the action of $T_{pol}$. The dilation $L^* \mapsto L_c$ (via the exponential time dilation $\tau = e^{\frac{1}{2} t}$) generates an equivalent orbit starting from $L_c$, which turns out to be the family of exp-Nesterov Lagrangians (6.1).

Concretely, we define the subgroup of linear time dilations $T_{lin} \subseteq T$:

$$T_{lin} = \{ \theta_c(t) = ct : c > 0 \} \quad (7.8)$$

---

15 Explicitly, if $\alpha^* = - \log t + \log 2$ is the scale function for $L^*$, then we can check that $(\alpha^*)^{(\tau)} = \alpha^*(\tau) + \log \dot{\tau} = \alpha$. 

32
which is isomorphic to $T_{\text{pol}}$. The discussion above says that the family of exp-Nesterov Lagrangians is an orbit under the action of $T_{\text{lin}}$. This means, in a precise sense, the Nesterov and exp-Nesterov families are isomorphic. Moreover, we can speed up Nesterov Lagrangian to get exp-Nesterov Lagrangian via an exponential time dilation function $\tau$. Furthermore, the associativity of the group action means we can speed up time to go from Nesterov to exp-Nesterov and back, and the results will remain consistent.

We can summarize our discussion by saying that all triangles in the following diagram commute (and we can reverse any arrow by replacing $\tau$ with $\tau^{-1}$):

![Diagram showing the commutative triangles between Nesterov and exp-Nesterov families, with associated equations for $\tau$, $p$, $q$, $r$, $\alpha_i$, $\rho_i$, and $\log t$.]
A Natural gradient descent and mirror descent

We review the equivalence between natural gradient descent and mirror descent.

A.1 Natural gradient descent and natural gradient flow

Natural gradient descent. We can interpret natural gradient descent as the solution to a modified optimization problem, where we now measure the norm of the displacement using the Hessian metric:

\[ x_{k+1} = x_k + v_k \]

\[ v_k = \arg \min_v \left\{ f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{\epsilon} \cdot \frac{1}{2} \|v\|^2_{h(x_k)} \right\}. \] (A.1a)

This means at each point \( x \) in \( X \) we have a local inner product and norm:

\[ \langle u, v \rangle_{h(x)} = \langle u, \nabla^2 h(x)v \rangle, \quad \|v\|^2_{h(x)} = \langle v, v \rangle_{h(x)} = \langle v, \nabla^2 h(x)v \rangle. \] (A.2)

Natural gradient flow. The continuous time limit (\( \epsilon \to 0 \) with time scaling \( t = \epsilon k \)) of natural gradient descent (A.1) which we call natural gradient flow is simply gradient flow on this space:

\[ \dot{X}_t = -\nabla^2 h(X_t)^{-1} \nabla f(X_t) \] (A.3)

Convergence of Natural Gradient Flow Define the energy functional

\[ E_t = t(f(X_t) - f^*) + D_h(x^*, X_t) \]

where \( x^* = \arg \min_x f(x) \) and \( f^* = f(x^*) \). It has time derivative:

\[ \dot{E}_t = f(X_t) - f^* + t(\nabla f(X_t), \dot{X}_t) - \left\langle \frac{d}{dt} \nabla h(X_t), x^* - X_t \right\rangle \]

Therfore, if \( X_t \) is governed by natural gradient flow (A.3), \( \dot{E}_t \) simplifies to

\[ \dot{E}_t = (f(X_t) - f^* + \langle \nabla f(X_t), x^* - X_t \rangle) + t(\nabla f(X_t), \dot{X}_t) \leq t(\nabla f(X_t), \dot{X}_t) \leq 0 \] (A.4)

using the convexity of \( f \). The energy is therefore decreasing over time: \( E_t \leq E_0 = D(x^*, X_0), t \geq 0 \). Thus, we conclude that natural gradient flow (A.3) has convergence guarantee:

\[ f(X_t) - f^* \leq \frac{D_h(x^*, X_0)}{t} = O \left( \frac{1}{t} \right). \]
A.2 Mirror descent and mirror flow

**Mirror descent.** The link between natural gradient descent and mirror descent is given by the Bregman divergence. The Bregman divergence, being a first order approximation error, has the property that it approximates the Hessian norm (A.2) without requiring the second order derivative:

\[ D_h(y, x) \approx \frac{1}{2} \langle y - x, \nabla^2 h(x)(y - x) \rangle = \frac{1}{2} \| y - x \|^2_{h(x)}. \]  
(A.5)

If we replace the Hessian norm in the optimization problem (A.1a) defining natural gradient descent, then we obtain mirror descent:

\[ x_{k+1} = x_k + v_k \]

\[ v_k = \arg \min_v \left\{ f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{\epsilon} D_h(x_k + v, x_k) \right\}. \]  
(A.6)

Setting the derivative of (A.6) to zero, we can also write mirror descent explicitly as:

\[ \nabla h(x_{k+1}) = \nabla h(x_k) - \epsilon \nabla f(x_k). \]  
(A.7)

**Mirror flow.** The continuous time limit (\( \epsilon \to 0 \) with time scaling \( t = \epsilon k \)) of mirror descent (A.7) is the mirror flow, which is the system:

\[ Z_t = \nabla h(X_t) \]  
(A.8a)

\[ \dot{Z}_t = -\nabla f(X_t) \]  
(A.8b)

Therefore, mirror flow (A.8) is equivalent to natural gradient flow (A.3):

\[ \frac{d}{dt} \nabla h(X_t) = \dot{Z}_t = -\nabla f(X_t) \iff \dot{X}_t = -\nabla^2 h(X_t)^{-1} \nabla f(X_t). \]  
(A.9)

Furthermore, mirror flow still has the same \( O(1/t) \) convergence in continuous time for any convex function \( f \). While mirror descent and mirror flow have matching convergence rates, it is difficult to prove any convergence rate for natural gradient descent; all we can say is that it is a descent method if \( f \) is smooth with respect to \( \| \cdot \|_{h(x)} \).

A.3 Natural gradient flow and mirror flow equivalence

In (A.9) we showed an “algebraic trick” that shows how the same differential equation can be written in two different ways, demonstrating the equivalence between mirror flow and (natural) gradient flow. Formally, we can understand this trick as the manifestation of the following property: mirror flow is the pushforward of natural gradient flow under the mapping \( \Phi = \nabla h \). In particular, mirror flow is also a gradient flow in the “dual manifold” \( Z = \nabla h(X) \). To illustrate this property, we show how the gradient flow changes when we transform the space.
**Mapping the space.** Suppose we map \( X \) to \( Z = \Phi(X) \) by a bijective smooth map:

\[
\Phi: X \to Z
\]

with inverse map (also smooth) \( \Psi = \Phi^{-1}: Z \to X \). The objective function \( f: X \to \mathbb{R} \) transforms to a new objective function \( \tilde{f}: Z \to \mathbb{R} \) given by:

\[
\tilde{f} = f \circ \Psi
\]

so that if \( z = \Phi(x) \):

\[
\tilde{f}(z) = \tilde{f}(\Phi(x)) = f(x).
\] (A.10)

However, note that \( \tilde{f} \) is not necessarily convex (in \( z \)), even if \( f \) is (in \( x \)).

**How the gradient changes.** We will use the following notation:

\[
\partial_x f(x_0, z_0) = \left. \frac{\partial f(x, z)}{\partial x} \right|_{(x, z) = (x_0, z_0)}
\]

The new objective function \( \tilde{f} \) now has gradient at \( z_0 \) (in the new space \( Z \)):

\[
\nabla \tilde{f}(z_0) = \partial_z \tilde{f}(z_0) = \partial_z (f \circ \Psi)(z_0) = J_{\Psi}(z_0) \partial_x f(\Psi(z_0)) = J_{\Psi}(z_0) \nabla f(\Psi(z_0))
\] (A.11)

where \( J_{\Psi}(z_0) \) is the Jacobian (partial derivatives) of \( \Psi(z) \) at \( z = z_0 \), represented as a matrix:

\[
(J_{\Psi}(z_0))_{ij} = \partial_i(\Psi(z_0))_j = \left. \frac{\partial \Psi_j(z)}{\partial z_i} \right|_{z = z_0}.
\] (A.12)

**How the metric changes.** Suppose \( X \) has metric \( g(x) \) at point \( x \) (e.g., Hessian metric \( g = \nabla^2 h \)). Then we can obtain a corresponding metric in \( Z = \Phi(X) \) which is the pullback metric of \( g \) under the inverse mapping \( \Psi = \Phi^{-1} \):

\[
\tilde{g} = \Psi^* g.
\]

Explicitly, this means the inner product at the point \( z_0 = \Phi(x_0) \) is given by:

\[
\langle u, v \rangle_{z_0} = \langle J_{\Psi}(z_0)u, J_{\Psi}(z_0)v \rangle_{x_0} = \langle J_{\Psi}(z_0)u, g(x_0)J_{\Psi}(z_0)v \rangle.
\]

That is, the metric \( g(x_0) \) at \( x_0 = \Psi(z_0) \) now becomes the metric \( \tilde{g}(z_0) \) at \( z_0 \):

\[
\tilde{g}(z_0) = (\Psi^* g)(z_0) = J_{\Psi}(z_0)^\top (g \circ \Psi)(z_0) J_{\Psi}(z_0).
\] (A.13)
How the gradient flow changes. Suppose in the original space $\mathcal{X}$ we have gradient flow, which (with the general definition of metric $g$) is given by:

$$
\dot{X}_t = -g(X_t)^{-1}\nabla f(X_t).
$$

In the new space $\mathcal{Z} = \Phi(\mathcal{X})$ with the pushforward metric $\tilde{g} = \Psi^*g$ and the new objective function $\tilde{f} = f \circ \Psi$, the natural gradient flow equation (A.3) becomes:

$$
\dot{Z}_t = -\tilde{g}(Z_t)^{-1}\nabla \tilde{f}(Z_t).
$$

Plugging in (A.13) then gives us:

$$
\dot{Z}_t = -[J_\Psi(Z_t)^\top g(\Psi(Z_t)) J_\Psi(Z_t)]^{-1} J_\Psi(Z_t) \nabla f(\Psi(Z_t))
= -J_\Psi(Z_t)^{-1} g(\Psi(Z_t))^{-1} \nabla f(\Psi(Z_t)).
$$

(A.14)

Mirror map from Legendre duality. Now consider the Hessian metric again:

$$
g = \nabla^2 h.
$$

In this case, there is a very nice choice of $\Psi$, called the mirror map:

$$
\Psi = \nabla h^*.
$$

Here $h^*: \mathcal{X}^* \to \mathcal{X}$ is the Legendre dual function, defined on the space of linear functionals $\mathcal{X}^*$:

$$
h^*(z) = \sup_x \langle z, x \rangle - h(x). \tag{A.15}
$$

The optimum in (A.15) is achieved by $x$ satisfying $z = \nabla h(x)$. So for all $x \in \mathcal{X}$, we have the relation:

$$
h^*(\nabla h(x)) = \langle \nabla h(x), x \rangle - h(x). \tag{A.16}
$$

Similarly, since $(h^*)^* = h$, for all $z \in \mathcal{Z}$ we also have:

$$
h(\nabla h^*(z)) = \langle \nabla h^*(z), z \rangle - h^*(z). \tag{A.17}
$$

Comparing (A.16) and (A.17), we conclude:

$$
z = \nabla h(x) \iff x = \nabla h^*(z)
$$

which means $\nabla h^* = (\nabla h)^{-1}$, so for all $z \in \mathcal{Z}$:

$$
\nabla h(\nabla h^*(z)) = z.
$$

Differentiating (calculating the Jacobian with respect to $z$) of the expression above gives us:

$$
\nabla^2 h^*(z) \nabla^2 h(\nabla h^*(z)) = I.
$$
So with the Hessian metric \( g = \nabla^2 h \) and the mirror map \( \Psi = \nabla h^* (\Phi = \nabla h) \), we have:

\[
J_{\Psi}(Z_t) = \partial_z \nabla h^*(Z_t) = \nabla^2 h^*(Z_t)
\]

\[
g(\Psi(Z_t)) = \nabla^2 h(\nabla h^*(Z_t)) = (\nabla^2 h^*(Z_t))^{-1}
\]

Notice how the choice of the mirror map makes \( J_{\Psi}(Z_t) \) and \( g(\Psi(Z_t)) \) cancel each other. Therefore, the pushforward of the natural gradient flow (A.14) in \( Z \) is indeed the mirror flow (A.8):

\[
\dot{Z}_t = -\nabla f(X_t)
\]

### **A.4 Natural gradient flow as massless limit of Hessian Lagrangian flow**

We consider the damped Lagrangian using the Hessian metric:

\[
\mathcal{L}(X_t, \dot{X}_t, t) = e^{\gamma t} \left( \frac{m}{2} ||\dot{X}_t||^2_{h(X_t)} - f(X_t) \right)
\]  

(A.18)

The Euler-Lagrange equation corresponding to (A.18) is:

\[
\frac{m}{2} \nabla^3 h(X_t) \dot{X}_t \dot{X}_t + \nabla^2 h(X_t) (m \ddot{X}_t + m \gamma_t \dot{X}_t) + \nabla f(X_t) = 0
\]  

(A.19)

when \( \gamma_t = t/m \), then in the limit \( m \to 0 \), the equation (A.19) converges to the first order equation

\[
\nabla^2 h(X_t) \dot{X}_t + \nabla f(X_t) = 0
\]

which is equivalent to the natural gradient flow (A.3)

### **B Deferred proofs**

#### **B.1 Proof of Lemma 1**

We prove Lemma 1, following the technique of [9, Theorem 1]. Since \( f \) is \( \frac{(p-1)!}{\epsilon} \)-smooth of order \( p-1 \), and we define \( x_{k+1} \) by (2.7), we have the following bound:

\[
f(x_{k+1}) \leq \min_x \left\{ f(x) + \frac{2}{\epsilon} \cdot \frac{1}{p} ||x - x_k||^p \right\}.
\]  

(B.1)

Moreover, choosing \( v = x^* - x_k \) in (B.1) gives us the bound:

\[
f(x_{k+1}) - f^* \leq \frac{2}{\epsilon} \cdot \frac{1}{p} ||x^* - x_k||^p.
\]

For any \( \lambda \in (0, 1) \), consider the midpoint:

\[
x_\lambda = x^* + (1 - \lambda)(x_k - x^*) = \lambda x^* + (1 - \lambda)x_k.
\]

Then from the convexity of \( f \) (1.12), we have the bound \( f(x_\lambda) \leq \lambda f^* + (1 - \lambda)f(x_k) \). Plugging this to (B.1) with the choice \( v = x_\lambda - x_k \), we find:

\[
f(x_{k+1}) \leq f(x_\lambda) + \frac{2}{\epsilon} \cdot \frac{1}{p} ||x_\lambda - x_k||^p \leq \lambda f^* + (1 - \lambda)f(x_k) + \frac{2}{\epsilon} \cdot \frac{1}{p} R^p \lambda^p.
\]
Then with our notation $\delta_k = f(x_k) - f^*$, we can write the last inequality above more precisely as:

$$\delta_{k+1} \leq (1 - \lambda)\delta_k + \frac{2}{\epsilon} \cdot \frac{1}{p} R^p \lambda^p.$$ 

Minimizing the right hand side with respect to $\lambda$ yields the optimal bound:

$$\delta_{k+1} \leq \delta_k - \frac{(p - 1)}{p} \cdot \left( \frac{\epsilon \delta_k^p}{2R^p} \right)^{\frac{1}{p-1}}.$$  \hspace{1cm} (B.2)

Now consider the energy functional $e_k = \delta_k^{-\frac{1}{p-1}}$; we have:

$$e_{k+1} - e_k = \frac{1}{\delta_{k+1}^{\frac{1}{p-1}}} - \frac{1}{\delta_k^{\frac{1}{p-1}}} = \frac{\delta_k^{\frac{1}{p-1}} - \delta_{k+1}^{\frac{1}{p-1}}}{\delta_k^{\frac{1}{p-1}} \cdot \delta_{k+1}^{\frac{1}{p-1}}} = \frac{\delta_k - \delta_{k+1}}{\delta_k^{\frac{1}{p-1}} \cdot \delta_{k+1}^{\frac{1}{p-1}}} \cdot \frac{1}{\left( \sum_{i=0}^{p-2} \delta_k^{\frac{1}{p-1}} \cdot \delta_{k+1}^{\frac{p-2-i}{p-1}} \right)}.$$  \hspace{1cm} (B.3)

We know that $G_p$ is a descent method, so $e_{k+1} \geq e_k$. The summation in the denominator of (B.3) can be bounded above by $(p - 1)\delta_k^\frac{p-2}{p-1}$, and we can lower bound $\delta_k - \delta_{k-1}$ using (B.2). Therefore:

$$e_{k+1} - e_k \geq \frac{(p - 1)}{p} \cdot \left( \frac{\epsilon \delta_k^p}{2R^p} \right)^{\frac{1}{p-1}} \cdot \frac{1}{\delta_k^{\frac{1}{p-1}}} \cdot \frac{1}{(p - 1)\delta_k^{\frac{p-2}{p-1}}} = \frac{1}{p} \cdot \left( \frac{\epsilon}{2R^p} \right)^{\frac{1}{p-1}}.$$  \hspace{1cm} (B.4)

Summing (B.4) and telescoping the terms, we get:

$$e_k \geq e_k - e_0 \geq \frac{k}{p} \cdot \left( \frac{\epsilon}{2R^p} \right)^{\frac{1}{p-1}}$$

which gives us the conclusion. \hspace{1cm} \footnote{\hspace{1cm}}

**B.2 Proof of Theorem 3**

Consider the energy functional (2.14):

$$E_t = \left( f(X_t) - f^* \right)^{-\frac{1}{p-1}}.$$  \hspace{1cm} (B.5)

Its time derivative is:

$$\dot{E}_t = -\frac{1}{(p - 1)} \cdot \frac{\langle \nabla f(X_t), \dot{X}_t \rangle}{(f(X_t) - f^*)^{\frac{p}{p-1}}}.$$  \hspace{1cm}

If $X_t$ evolves following the rescaled gradient flow (2.12), then $\dot{E}_t$ simplifies to:

$$\dot{E}_t = \frac{1}{(p - 1)} \cdot \left( \frac{\|\nabla f(X_t)\|_s}{f(X_t) - f^*} \right)^{\frac{p}{p-1}}.$$  \hspace{1cm} (B.6)

Notice that by the convexity of $f$, we have

$$0 \leq f(X_t) - f^* \leq \langle \nabla f(X_t), X_t - x^* \rangle \leq \|\nabla f(X_t)\|_s \cdot \|X_t - x^*\|.$$
and therefore, from (B.6) we obtain a bound:
\[
\dot{E}_t \geq \frac{1}{(p-1)} \cdot \frac{1}{\|X_t - x^*\|^{\frac{p}{p-1}}}. 
\]
Integrating, this gives a lower bound on \(E_t\):
\[
E_t \geq E_0 + \frac{1}{(p-1)} \int_0^t \frac{1}{\|X_\tau - x^*\|^{\frac{p}{p-1}}} d\tau. \tag{B.7}
\]
Furthermore, under the additional mild assumption that the level sets are bounded, the bound (B.7) simplifies to:
\[
E_t \geq E_0 + \frac{t}{(p-1) R^{\frac{p}{p-1}}}. 
\]
Now recalling the definition (B.5), this lower bound becomes an upper bound on the function values:
\[
f(X_t) - f^* \leq \left( E_0 + \frac{t}{(p-1) R^{\frac{p}{p-1}}} \right)^{-(p-1)}. 
\]
Finally, replacing \(E_0 \geq 0\) by 0 completes the proof.

B.3 Proof of Lemma 4

We follow the proof of [9, Lemma 6]. Since \(y_k\) solves the optimization problem (3.2a), it satisfies the optimality condition:
\[
\sum_{i=1}^{p-1} \frac{1}{(i-1)!} \nabla^i f(x_k) (y_k - x_k)^{i-1} + \frac{2}{\epsilon} \|y_k - x_k\|^{p-2} (y_k - x_k) = 0. \tag{B.8}
\]
Furthermore, since \(\nabla^{p-1} f\) is \(\frac{(p-1)!}{\epsilon}\)-Lipschitz, we have the following error bound on the \((p-2)\)-nd order Taylor expansion of \(\nabla f\):
\[
\left\| \nabla f(y_k) - \sum_{i=1}^{p-1} \frac{1}{(i-1)!} \nabla^i f(x_k) (y_k - x_k)^{i-1} \right\|_\ast \leq \frac{1}{\epsilon} \|y_k - x_k\|^{p-1}. \tag{B.9}
\]
Substituting (B.8) to the square of (B.9) and writing \(r = \|y_k - x_k\|\), we obtain:
\[
\frac{r^{2p-2}}{\epsilon^2} \geq \left\| \nabla f(y_k) + \frac{2r^{p-2}}{\epsilon} (y_k - x_k) \right\|_\ast^2. 
\]
Upon expanding the square and rearranging, we get the inequality:
\[
\langle \nabla f(y_k), x_k - y_k \rangle \geq \frac{\epsilon}{4 r^{p-2}} \|\nabla f(y_k)\|_\ast^2 + \frac{3r^p}{4\epsilon}. \tag{B.10}
\]
Note that if $p = 2$, then the first term in (B.10) above already implies the desired bound (3.5). Now assume $p \geq 3$. The right hand side of (B.10) is a convex function of $r$, and it is minimized by $r^* = \{ \frac{p-2}{3p} \epsilon^2 \| \nabla f(y_k) \|_2 \} \frac{1}{p-2}$, yielding a lower bound of (B.10) that is now independent of $r$:

$$\langle \nabla f(y_k), x_k - y_k \rangle \geq \frac{1}{4} \left( \left( \frac{3p}{p-2} \right)^{\frac{p-2}{2p-2}} + \left( \frac{p - 2}{3p} \right)^{\frac{p}{2p-2}} \right) \epsilon^{\frac{1}{p-1}} \| \nabla f(y_k) \|_2^{\frac{p}{p-1}}$$

$$\geq \frac{1}{4} \epsilon^{\frac{1}{p-1}} \| \nabla f(y_k) \|_2^{\frac{p}{p-1}}$$

as desired. \qed
References

[1] Zeyuan Allen-Zhu and Lorenzo Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. *arXiv preprint arXiv:1407.1537*, 2014.

[2] Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, February 1998.

[3] Michel Baes. Estimate sequence methods: Extensions and approximations. Manuscript, available at [http://www.optimization-online.org/DB_FILE/2009/08/2372.pdf](http://www.optimization-online.org/DB_FILE/2009/08/2372.pdf), August 2009.

[4] Israel M. Gelfand and Sergei V. Fomin. *Calculus of Variations*. Dover Books on Mathematics Series. Dover Publications, 2000.

[5] Kenneth Lange and Kevin L. Keys. The proximal distance algorithm. In *Proceedings of the International Congress of Mathematicians*, volume IV, pages 95–116, Seoul, 2014.

[6] Yuriii Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[7] Yuriii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization. Kluwer, Boston, 2004.

[8] Yuriii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[9] Yuriii Nesterov. Accelerating the cubic regularization of Newton’s method on convex problems. *Mathematical Programming*, 112(1):159–181, 2008.

[10] Yuriii Nesterov and Boris T. Polyak. Cubic regularization of Newton’s method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[11] Brendan O’Donoghue and Emmanuel Candès. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, 15(3):715–732, 2015.

[12] Garvesh Raskutti and Sayan Mukherjee. The information geometry of mirror descent. *IEEE Transactions on Information Theory*, 61(3):1451–1457, 2015.

[13] Weijie Su, Stephen Boyd, and Emmanuel Candès. A differential equation for modeling Nesterov’s accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems (NIPS) 27*, 2014.