Acceleration in optimization: perspective from geometry and classical physics

Alisher Mirmanov

November 2023

Abstract

This project serves as a literature review on physical and geometrical interpretations of a popular modern optimization technique, Nesterov Gradient Descent (NGD), developed by Yurii Nesterov in [10], which has been shown to achieve a quadratic convergence rate. After defining the setting of the investigation, we state and compare NGD with conventional Gradient Descent (GD). Further, in order to better understand its accelerated convergence rate, the continuous time limit of the algorithm is derived, following the work proposed in [3] and [14]. The system is analyzed under the calculus of variation framework, showing the exponential convergence rate of $O(\frac{1}{e^{\beta}})$ for the ODE. Additionally, as in [15], by focusing on a specific β - smooth quadratic function, certain conditions are imposed on the ODE, for it to truly optimize action. For practical application, a general form of discretization of rate-matching Nesterov ODE is presented, with an arbitrary convergence rate of $O(\frac{1}{k^p})$ in normed spaces, a special case of which is in fact NGD. To offer insights into Nesterov optimization in a non-Euclidean setting, the extension of all these results in a manifolds setting is considered, first developed in [1] and later by [5], showing that the ODE achieves exponential convergence rate for geodesically convex functions, and extending the discretization of the ODE. Lastly, the penalty function approximation problem and Batch Normalization problem are used for illustration, which could be solved using the methods explored in the paper.

Table of Contents

1 Introduction

Optimization plays a very important role in a variety of scientific and engineering problems. The applications include problems in machine learning, medical imaging, physics, etc. To address these problems, a variety of optimization methods have been developed, including the first gradient descent algorithm introduced by Cauchy in 1847 to simulate the orbits of celestial bodies. Although traditional optimization methods were foundational in many applications, more sophisticated optimization methods were constructed, that allowed for a faster convergence rate. One such algorithm is the Nesterov Gradient Descent presented by Yuri Nesterov in [10], which introduced a momentum term into the algorithm allowing for faster convergence of $O(\frac{1}{k^2})$. Later, other accelerated methods were developed like accelerated mirror descent and accelerated cubic-regularized Newton's method, but these are not part of the investigation.

In an attempt to better understand NGD, and get insight into its quadratic convergence, Nesterov's ODE was derived in [3], by taking a limit to 0 in the step size of the algorithm. This allowed for a deeper analysis of the problem, by considering it through a variational framework using a family of time-dependent Bregman Lagrangian. Some interesting properties of the ODE are thoroughly studied, and it has been shown that the solution to the ODE converges to its minimizer with exponential convergence rate, $O(\frac{1}{e^\beta})$. For all practical purposes, general form discretizations of the ODE are derived in [14], that could be accelerated to arbitrary precision of $O(\frac{1}{k^p})$, and for which NGD is a special case of rate – matching discretization. However, in deeper analysis of the action functional in [15], it has been shown that the Nesterov ODE doesn't always extremize the action, and in general, there are conditions discussed in Chapter 5 to be imposed on the time interval and the action J for the ODE to extremize it. To demonstrate, a specific β - smooth quadratic function is used.

The applicability of these findings is further expanded, by generalizing the results achieved in [14] into the manifold setting in [1] and [5]. It has been shown that the ODE can achieve an exponential convergence rate, and the discretization proposed in [5] can also be accelerated to arbitrary polynomial precision, effectively extending previous results into a new setting.

Finally, these findings have their applications in a vast class of problems, like Semi-Definite programming, distance minimization, eigenvalue problems, etc. In this paper, we consider 2 of such problems, namely, the penalty function approximation problem and Batch Normalization problems, without directly solving them.

2 Convex functions

Before one begins investigation of optimization methods it is important to understand the setting in which it's carried out. Specific choice of families of functions, that would be subject to optimization are presented. This choice is mainly inspired by [3], [12]. Hence, this section mainly focuses on

so called α - strongly convex functions, that are also $L - smooth$, providing important definitions and properties of such functions, that are used later in the work. One, however, starts by defining a convex set and convex function.

2.1 Convex sets and convex functions

Definition 2.1. *A set C is convex if for any two points in C, the line segment between them is also inside C. Formally, $\forall c_1, c_2 \in C$ and any $\lambda \in [0, 1]$, we get that:*

$$\lambda c_1 + (1 - \lambda)c_2 \in C.$$

Definition 2.2. *A function $f : \mathbb{R}^d \to \mathbb{R}$ is called convex if $\forall x, y \in \mathbb{R}^d$ it satisfies the following property:*

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y), \quad \lambda \in [0, 1].$$

From this definition, one immediately gets that for any convex function f, it's linear Taylor approximation at any point is below the said function. Thus, one gets the following property for the convex functions:

A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if $\forall x, y \in \mathbb{R}^d$:

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

where the inner product is the standard Eucledian inner product.

Using this property, it's not hard to see that the Hessian of the convex function, defined by is positive semi-definite, i.e.: A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if $\forall x, y \in \mathbb{R}^d$:

$$\nabla^2 f(x) \quad \text{is positive semi-definite, i.e. all eigenvalues of} \quad \nabla^2 f(x) \geq 0.$$

Note, however,that we require the function to be twice differentiable, unlike Definition 2.2, and that for a special case when $n = 1$, we get that $f^{(2)}(x) \geq 0$.

In order to avoid mathematical obstacles when analyzing the convergence rates and deriving differential equation for NGD, one has to apply some constraints on the family of convex functions, by considering the following definitions.

Definition 2.3. *A function $f : \mathbb{R}^d \to \mathbb{R}$, differentiable, is called L-smooth if its gradient is Lipschitz continious with parameter L, i.e.: $\forall x, y \in \mathbb{R}^n$ it satisfies the following property:*

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

The family of convex functions with L–Lipschitz continuous gradients is denoted by $\mathcal{F}_l$.

The following immediate consequence from this definition could be stated as a theorem:

Theorem 1. *Let f be L-smooth function, the the function: $\frac{L}{2}\|x\|^2 - f(x)$ is convex.*

Proof. Let $g(x) = \frac{L}{2}\|x\|^2 - f(x)$. Then:

$$\begin{aligned}\langle \nabla g(x) - \nabla g(y), x - y\rangle &= \langle L(x-y) - (\nabla f(x) - \nabla f(y)), x - y\rangle \\ &= L\|x-y\|^2 - \langle \nabla f(x) - \nabla f(y), x - y\rangle \\ &\geq L\|x-y\|^2 - \|\nabla f(x) - \nabla f(y)\|^2\|x-y\|^2 \\ &\geq L\|x-y\|^2 - L\|x-y\|\|x-y\| = 0,\end{aligned}$$

where the definition of L-smooth functions and Cauchy-Schwartz inequality[1] were used. Hence, one gets that $\nabla g(x)$ is monotone, and given continuity and differentiability of $g(x)$, $g(x)$ is convex.[2] $\square$

Using the fact that $\frac{L}{2}\|x\|^2 - f(x)$ is convex one can obtain an upper bound on the function $f(y)$:

$$\begin{aligned}\frac{L}{2}\|y\|^2 - f(y) &\geq \frac{L}{2}\|x\|^2 - f(x) + \langle Lx - \nabla f(x), y - x\rangle \\ f(y) &\leq f(x) + \langle \nabla f(x), y - x\rangle - L\langle x, y\rangle - \frac{L}{2}\|y\|^2 + L\|x\|^2 \\ &= f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}\|y - x\|^2.\end{aligned}$$

Definition 2.4. *A function $f : \mathbb{R}^d \to \mathbb{R}$ is called α-strongly convex if $g(x) = f(x) - \frac{\alpha}{2}\|x\|^2$ is convex.*

By substituting into the previous definition of the convex function, we get the following property for the α-strongly convex functions:

$$f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle + \frac{\alpha}{2}\|y - x\|^2. \tag{1}$$

3 Optimization methods

By considering definitions made in the previous section, one can proceed to look at different optimization methods. The objective of any optimizaiton problem is to find a point $x^* \in \mathbb{R}^d$, minimizer, that minimizes the objective function $f(x)$. Namely, one needs to solve

$$\min_{x \in \mathbb{R}^d} f(x),$$

for some $L - smooth\ f \in \mathcal{C}^1(\mathbb{R}^d, \mathbb{R})$.

The most foundational optimization algorithm is gradient descent algorithm (GD), with a given convergence rate $O\left(\frac{1}{k}\right)$. A more sophisticated scheme is the Accelerated Gradient Descent or Nesterov Gradeint Descent(NGD), that minimizes at rate $O\left(\frac{1}{k^2}\right)$.

3.1 Gradient Descent algorithm

The conventional gradient descent(GD) algorithm was first introduced by Augustin-Louis Cauchy in 1847 to describe the systems of equations related to the planetary motion. The formalism

[1] $|\langle \mathbf{u}, \mathbf{v}\rangle| \leq \|\mathbf{u}\| \cdot \|\mathbf{v}\|$

[2] Proof in [2]

iteratively approaches a minimum of the object function by moving in the opposite direction of the gradient of the function, the direction of the steepest increase of the function, with step size η:

$$x_k = x_{k-1} - \eta \nabla f(x_{k-1}). \tag{2}$$

3.1.1 Convergence rate for the Gradient Descent

Using this formalism, one can identify the optimal value of η in the context of $L - smooth$ and α - strongly convex functions. In particular, it's not hard to see that $\eta = \frac{1}{L}$ is the simplest case for optimal convergence of the algorithm. Let x^+ be immediate iteration after x, then

$$x^+ - x = \eta \nabla f(x)$$

By using the fact that the subject function is $L - smooth$, we get that:

$$\begin{aligned} f(x^+) &\leq f(x) + \langle \nabla f(x), x^+ - x \rangle + \frac{L}{2}\|x^+ - x\|^2 \\ &= f(x) - \eta \langle \nabla f(x), \nabla f(x) \rangle + \frac{L}{2}\|x^+ - x\|^2 \\ &= f(x) - \eta \|\nabla f(x)\|^2 + \frac{L}{2}\|x^+ - x\|^2 \\ &= f(x) - \eta \left(1 - \frac{L}{2}\eta\right) \|\nabla f(x)\|^2. \end{aligned}$$

Using the property of convex functions that $\|\nabla f(x)\|^2 \geq 0$. As a result one needs to choose η small enough so that it can guarantees the improvement in the algorithm, i.e. make the inequality above strict . In particular, we need to set $\eta \left(1 - \frac{L}{2}\eta\right) > 0$. Thus $\eta \leq \frac{1}{L}$

Using $\|\nabla f(x)\|^2$, one can workout the convergence rate of the gradient descent, i.e. obtain the bounds on the suboptimality of x_k:

$$|f(x_k) - f(x^*)| \leq O\left(\frac{1}{k}\right)$$

In particular we have that by convexity and Lipschitz continuity of f:

$$f(x_k) - f^* \leq \langle \nabla f(x_k), x_k - x^* \rangle \leq L\|x_k - x^*\|^2.$$

Summing over $k = 0, 1, \ldots, T-1$ and dividing both sides by T yields

$$\frac{1}{T}\sum_{k=0}^{T-1}(f(x_k) - f^*) \leq \frac{L}{T}\sum_{k=0}^{T-1}\|x_k - x^*\|^2.$$

By the convexity of f, we have

$$f\left(\frac{1}{T}\sum_{k=0}^{T-1} x_k\right) - f^* \leq \frac{1}{T}\sum_{k=0}^{T-1}(f(x_k) - f^*).$$

Using the update rule for gradient descent $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$, we can rewrite the left-hand side as

$$f\left(\frac{1}{T}\sum_{k=0}^{T-1} x_k\right) - f^* = f(x_T) - f^* \quad \Longrightarrow \quad f(x_T) - f^* \leq \frac{L}{T}\sum_{k=0}^{T-1}\|x_k - x^*\|^2.$$

Choosing $T = k + 1$, we have

$$\boxed{f(x_k) - f^* \leq \frac{2L\|x_0 - x^*\|^2}{k+1}.}$$

3.2 Nesterov Gradient Descent

In 1983, Russian mathematician Yuri Nesterov, has proposed a new method of optimization that would ensure the convergence rate of $O\left(\frac{1}{k^2}\right)$ compared to $O\left(\frac{1}{k}\right)$ for GD [10]. NGD implemented the momentum term, first developed by Boris Polyak for an earlier optimization method [11].

The momentum term, incorporates some information about the direction of improvement at the previous iteration of the algorithm. As mentioned before, it is still quite unclear how momentum term allows for better convergence.

The NGD is described by the following algorithm

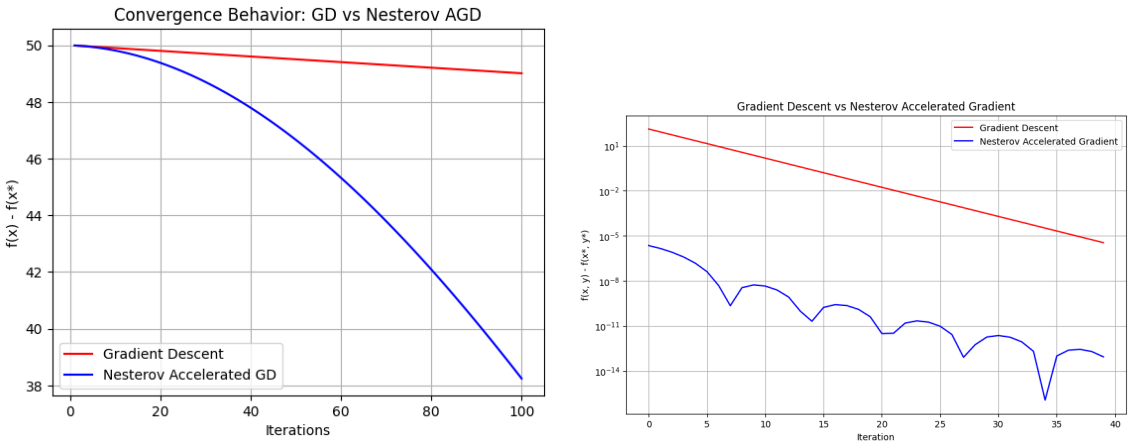
$$\begin{cases} x_k = y_{k-1} - \eta \nabla f(y_{k-1}), \\ y_k = x_k + \dfrac{k-1}{k+2}\left(x_k - x_{k-1}\right), \end{cases} \tag{3}$$

where y_k is precisely the momentum term. The proof of algorithms convergence rate is quite involved, and was shown in [10] to exhibit the quadratic convergence rate for smooth, strongly convex functions, i.e.:

$$\boxed{|f(x_k) - f(x^*)| \leq O\left(\frac{1}{k^2}\right),}$$

which is optimal among all methods implementing $\nabla f(x)$ at consecutive iterates, i.e. achieves the fastest convergence rate among algorithms that use the gradient of the function, which has been shows in [10].

A numerical result in Figure 1 compares the NGD to GD for a given covnex objective function.



(a) $f(x) = \frac{1}{2}x^2$. (b) $f(x, y) = x^2 + y^2$.

Figure 1: GD vs NGD convergence rate comparison.

4 Continuous time-limit for Nesterov Algorithm

To analyze Nesterov Descent though the lens of physics and geometry, i.e. as a mechanical system, one needs to derive the algorithm's continuous time-limit to find the differential equation the

solutions to which follow the path that the algorithms undertakes as it approaches the minimizer.

Thus, in this section the ODE that is associated to the algorithm is derived and some analysis is provided for better understanding of the movement.

4.1 Derivation of Continuous time-limit

Firstly, we combine the two expressions in (12), we get the following result:

$$\frac{x_{k+1}-x_k}{\sqrt{\eta}} = \frac{k-1}{k+2}\frac{x_k-x_{k-1}}{\sqrt{\eta}} - \sqrt{\eta}\nabla f(y_k). \tag{4}$$

For the function $f \in \mathcal{F}_l$ with $L > 0$, recall that η is the incremental step size for the algorithm. As one is looking for a continuous function $X(t)$, defined for $t \geq 0$, we require that $\eta = \Delta t = t_{k+1} - t_k$. Following the method described in [3], we introduce ansatz: $x_k \approx X(k\sqrt{\eta})$, i.e. the value of x at the k^{th} iteration is equal to the value of X at time $k\sqrt{\eta}$, where we put $k = \frac{t}{\sqrt{\eta}}$. This results in $X(t) \approx x_{t/\sqrt{\eta}} = x_k$ and $X(t+\sqrt{\eta}) \approx x_{(t+\sqrt{\eta})/\sqrt{\eta}} = x_{k+1}$.

To simplify the latter equation x_{k+1} is Taylor expanded to get:

$$x_{k+1} = X((k+1)\sqrt{\eta}) \approx X(k\sqrt{\eta}) + \sqrt{\eta}\dot{X}(k\sqrt{\eta}) + \frac{1}{2}\ddot{X}(k\sqrt{\eta})s + o(\sqrt{\eta^3}),$$

hence:

$$\begin{aligned}\frac{x_{k+1}-x_k}{\sqrt{\eta}} &= \frac{X(k\sqrt{\eta}) + \sqrt{\eta}\dot{X}(k\sqrt{\eta}) + \frac{1}{2}\ddot{X}(k\sqrt{\eta})s + o(\sqrt{\eta^3}) - X(k\sqrt{\eta})}{\sqrt{\eta}} \\ &= \dot{X}(k\sqrt{\eta}) + \frac{1}{2}\ddot{X}(k\sqrt{\eta})\sqrt{\eta} + o(\sqrt{\eta}) \\ &= \dot{X}(t) + \frac{1}{2}\ddot{X}(t)\sqrt{\eta} + o(\sqrt{\eta}),\end{aligned}$$

similarly, $\frac{x_{k+1}-x_k}{\sqrt{\eta}} = \dot{X}(t) - \frac{1}{2}\ddot{X}(t)\sqrt{\eta} + o(\sqrt{\eta})$ and $\sqrt{\eta}\nabla f(y_k) = \sqrt{\eta}\nabla f(X(t)) + o(\sqrt{\eta})$.

Thus, the equation (4) is rewritten as follows:

$$\dot{X}(t) + \frac{1}{2}\ddot{X}(t)\sqrt{\eta} + o(\sqrt{\eta}) = \left(1 - \frac{3\sqrt{\eta}}{t}\right)\left(\dot{X}(t) - \frac{1}{2}\ddot{X}(t)\sqrt{\eta} + o(\sqrt{\eta})\right) - \sqrt{\eta}\nabla f(X(t)) + o\sqrt{\eta}.$$

Comparing the coefficients in $\sqrt{\eta}$ one receives the following ODE describing the Nesterov Scheme:

$$\boxed{\ddot{X} + \frac{3}{t}\dot{X} + \nabla f(X) = 0.} \tag{5}$$

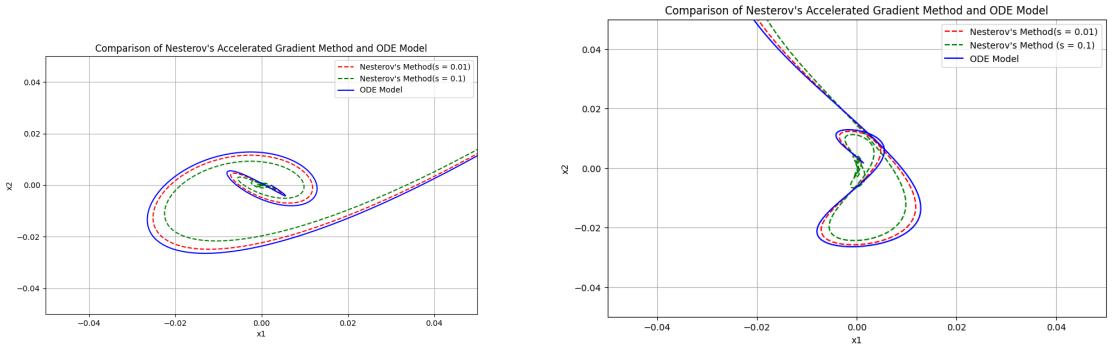
For $k = 1$, in (4), with the initial condition $X(0) = x_0$ results in $\dot{X}(0) = 0$ as $(x_2 - x_1)/\sqrt{\eta} = o(1)$.

An important observation is that the factor $\frac{3}{t}$ arises from $\frac{k-1}{k+2} \approx 1 - \frac{3}{k}$. In fact, by replacing $\frac{k-1}{k+2}$ term with more general $\frac{k-1}{k+r-1}$ term in Nesterov algorithm, one can derive a general ODE:

$$\ddot{X} + \frac{r}{t}\dot{X} + \nabla f(X) = 0, \tag{6}$$

for a constant r. However, the inverse quadratic convergence rate is only achieved if and only if $r \geq 3$, as stated in [12], where it was shown numerically that for $r < 3$ the convergence is inverse linear.

In Figure 2, NGD is compared to it's ODE on different test functions for step-size $\eta = s = 0.01$ and $s = 0.1$, proving the validity of the ODE, as s approaches 0.



(a) $f(x_1, x_2) = 2 * 10^{-2}x_1^2 + 3 * 10^{-3}x_2^2$.

(b) $f(x_1, x_2) = 2 * 10^{-2}x_1^2 + 0.5 * 10^{-3}x_2^2$.

Figure 2: GD vs NGD convergence rate comparison.

4.2 Properties of the ODE

By considering at the equation (5), some interesting properties emerge about the mechanics of the ODE.

Firstly, the **ODE is invariant under linear time transformation** $\tilde{t} = ct$, for some $c > 0$, as

$$\frac{dX}{d\tilde{t}} = \frac{1}{c}\frac{dX}{dt}, \frac{d^2X}{d\tilde{t}^2} = \frac{1}{c^2}\frac{d^2X}{dt^2},$$

yielding in the ODE:

$$\frac{d^2X}{d\tilde{t}^2} + \frac{3}{\tilde{t}}\frac{dX}{d\tilde{t}} + \frac{\nabla f(X)}{c^2} = 0,$$

for which the minimization of $f(X)$ and $kf(X)$ are equivalent for some constant k.

Additionally, if we regard $X(t)$ to be a smooth function, then by mean value theorem $\exists t^* \in (0, t)$ s.t. $\frac{\dot{X}(t)-\dot{X}(0)}{t} = \frac{\dot{X}(t)}{t} = \ddot{X}(t^*)$. This results in

$$\ddot{X}(t) + 3\ddot{X}(t^*) + \nabla f(X(t)) = 0.$$

Thus, as $t \to 0, \ddot{X}(0) = -\nabla f(x_0)/4$, describing, asymptotically, the behaviour as

$$X(t) = -\frac{\nabla f(x_0)t^2}{8} + x_0 + o(t^2).$$

One can clearly observe that for small values of t the algorithm exhibits slower motion.

5 Variational Perspective

The ODE derived in the previous section for the algorithm, allows one to analyze the motion of ODE from variational perspective. By considering the Lagrangian for the system, one can generate the large class of continuous accelerated algorithms, as well as extending it to a non - Euclidian setting.

In this section, I'm going to follow the work [14] and [15]. In particular, transition into a non-Euclidean space, by considering the Bregman Divergence as a distance metric and analyze the

Bregman Lagrangian that corresponds to the equation of motion described by the ODE. Using Bregman Lagrangian, it has been shown in [14] that in continuous setting we can achieve exponential convergence. One can further discuss conditions under which Nesterov truly minimizes the action functional.

5.1 Background on calculus of variations

For a given motion function, define a Lagrangian Function $L(X, \dot{X}, t)$. One can implement a notion of action functional, defined by

$$J[X] = \int_{t_1}^{t_2} L(X, \dot{X}, t)\, dt,$$

which takes a Lagrangian describing trajectory of motion as an argument, and is useful as minimizing action gives the optimal trajectory of motion. One can also make use of Euler - Lagrange equation, the solutions to which gives stationary points[3] of given action functional.

$$\frac{d}{dx}\left(\frac{\partial}{\partial \dot{X}} L(X, \dot{X}, t)\right) = \frac{\partial}{\partial X} L(X, \dot{X}, t).$$

For a given perturbation in the path $h = \delta X$, taken by the NGD, the derivative of J in direction h should be zero for all choices of h, to extrimise the action. We define the difference in the action potential as:

$$\Delta J[X, h] = J[X + h] - J[X]$$

equipping our space $\mathcal{C}([t_1, t_2], \mathbb{R}^d)$ with $\|X\| = \max_{t\in[t_1,t_2]} \|X\| + \max_{t\in[t_1,t_2]} \|\dot{X}\|$, described in more detail in [9] p.15, under which the action is continuous.

Like in [14] and [15], if the change in the action can be expressed as a linear term in h plus a small term, the functional is said to be differentiable, and the **first variation** is defined. Mathematically, $J[X, h] = \phi[X, h] + \epsilon\|h\|$ with $\epsilon \to 0$ as $\|h\| \to 0$. and the function ϕ is the first variation.

If the increment of the action can be expressed as a linear term (ϕ_1) and a quadratic term (ϕ_2) in h, i.e. $J[X, h] = \phi_1[X, h] + \phi_2[X, h] + \epsilon\|h\|^2$ the functional is said to be twice differentiable, i.e. second variation is defined: $\delta^2 J[Y, h] := \phi_2[X, h]$.

5.2 The Lagrangian for Nesterov ODE

Using the equaiton (5) one can develop a suitable Lagrangian that would satisfy the ODE. Since intrinsically $L = T - V$, where $T = \frac{1}{2}m\dot{X}^2$ is the kinetic energy of the system and V the potential energy, the first intuition is to use the simplest case for the Lagrangian:

$$L(X, \dot{X}, t) = t^3 \left(\frac{1}{2}\|\dot{X}\|^2 - f(X)\right),$$

[3]Paths for which the action is at a local minimum, maximum, or saddle point.

where the factor t^3 is introduced to include time dependence and make the equation satisfy the Euler - Lagrange equations. In fact, one gets:

$$\frac{d}{dx}\left(\frac{\partial}{\partial \dot{X}}L(X,\dot{X},t)\right) = t^3\ddot{X} + 3t^2\dot{X},$$
$$\frac{\partial}{\partial X}L(X,\dot{X},t) = t^3\nabla f(X),$$

so the Euler - Lagrange equation reduces the ODE for the NGD. In [14] this notion has been extended to non-Euclidean space $\mathcal{N}$, endowed with a convex distance generating function $\psi : \mathcal{N} \to \mathbb{R}$, by using the Bregman divergence defined as follows:

$$D_\psi(x,y) = \psi(x) - \psi(y) - \langle \psi(y), x - y\rangle$$

where inner product is defined by regular scalar product. Note that ψ has to be continuously differentiable for the distance function to be well - defined, and that the convexity of ψ ensures that the distance is non-negative. This formulation redefines the distance function for convex spaces, however, one can recover the norm for general Euclidean space by making $\psi = \frac{1}{2}\|x\|^2$.

Using Bregman divergence, the authors of [14] have introduced a non - Euclidian Lagrangian formulation to satisfy the NGD-ODE:

$$L_{\alpha,\beta,\gamma}(X,\dot{X},t) = fe^{\alpha_t+\gamma_t}\left(D_\psi(X + e^{-\alpha_t}V, X) - e^{\beta_t}f(X)\right)$$

where The functions $\alpha, \beta, \gamma : \mathbb{T} \to \mathbb{R}$ are the weighing functions for the velocity and the damping of the motion, and where $\mathbb{T} \subset \mathbb{R}$ is the time interval of the motion. Importantly, by taking $\psi(x) = \frac{1}{2}\|x\|^2$, and $\alpha = log(2/t), \beta = \gamma = 2log(t)$ we recover the initial Lagrangian from before. In [14] ideal scaling conditions were defined:

$$\dot{\beta}_t \leq e^{\alpha_t},$$
$$\dot{\gamma}_t = e^{\alpha_t}.$$

In this work, however, the justification for the conditions are omitted, for more detail refer to [14]. We note that the Euler - Lagrange equation imply that:

$$\ddot{X} + (e^{\alpha_t} - \dot{\alpha}_t)\dot{X} + e^{2\alpha_t+\beta_t}\left[\nabla^2\psi(X + e^{-\alpha_t}\dot{X})\right]^{-1}\nabla f(X) = 0, \tag{7}$$

which further simplifies

$$\frac{d}{dt}\nabla\psi\left(X_t + e^{-\alpha_t}\dot{X}_t\right) = -e^{\alpha_t+\beta t}\nabla f(X_t). \tag{8}$$

Under the ideal scaling conditions it has been shown in [14] that the solutions minimize the objective function at the exponential rate $O\left(\frac{1}{e^\beta}\right)$, stated in the following theorem.

Theorem 2. *Under the ideal scaling condition, convergence rate associated with solutions to the Euler-Lagrange equation satisfy:*

$$\boxed{|f(X_t) - f(x^*)| \leq O\left(\frac{1}{e^\beta}\right)}.$$

Proof. To prove, one uses the so-called Lyapunov (functions of constant energy) in accordance with [13] and [14]. Lyapunov function $\mathcal{V}$ is a continuous function $\mathcal{V} : \mathbb{R}^d \to \mathbb{R}$, that is non-negative, radially unbounded, zero at a fixed point and decreasing.

A Lyapunov function can be thought of as "energy". Define the following energy functional:

$$\mathcal{E}_t = D_\psi \left(x^*, X_t + e^{-\alpha_t}\dot{X}_t\right) + e^{\beta_t}(f(X_t) - f(x^*)),$$

then,

$$\dot{\mathcal{E}}_t = \langle \frac{d}{dt}\nabla\psi(X_t + e^{-\alpha_t}\dot{X}_t), x^* - X_t - e^{-\alpha_t}\dot{X}_t\rangle + \dot{\beta}_t e^{\beta_t}(f(X_t) - f(x^*))e^{\beta_t}\langle \nabla f(X_t), \dot{X}_t\rangle,$$

where if X satisfies (8):

$$\dot{\mathcal{E}}_t = -e^{\alpha_t+\beta_t}D_f(x^*, X_t) + (\dot{\beta}_t - e^{\alpha_t})e^{\beta_t}(f(X_t) - f(x^*)).$$

As Bregman divergence is non-negative implies that the first term in $\dot{\mathcal{E}}_t$ is non-positive. By ideal scaling condition, the second term is also non-positive which implies that $\dot{\mathcal{E}}_t \leq 0$.

Lastly, as $D_\psi(x^*, X_t) \geq 0$ for $t > t_0$ we get $e^{\beta_t}(f(X_t) - f(x^*)) \leq \mathcal{E}_t \leq \mathcal{E}_{t_0} \implies f(X_t) - f(x^*) \leq e^{\beta_t}\mathcal{E}_{t_0} = O\left(\frac{1}{e^{-\beta_t}}\right)$. □

This clearly shows that the derived ODE converges at exponential rate. In reality, however, one must consider the discretesation of the algorithm for it to be of practical use. This comes with great difficulty, and the discretesation of the sub-family of the Bregman Lagrangians were considered in [14]. In reality, a specific case of the rate matching discretesation, i.e. an algorithm with the mathcing convergence rate, turns out to be Nesterov Scheme, with the general form of rate matching discretesation presented in Appendix B.

5.3 Time Invariance

One intriguing property of this Lagrangian is the fact that it's closed under time reparametrization. In particular, let $\tau : \mathbb{T} \to \mathbb{T}' \subset \mathbb{R}$ be a smooth increasing function. For a curve, $X : \mathbb{T}' \to \mathcal{N}$, one can define $Y_t = X_{\tau(t)}$, $Y_t : \mathbb{T} \to \mathcal{N}$. For $\tau(t) > t$, Y_t is called the sped-up version of X, as Y_t at time t is the same as X at time $\tau(t) > t$. The general result, proved in [14] Appendix A1, the Lagrangian that Y_t satisfies transforms as follows.

$$L_{\tilde{\alpha},\tilde{\beta},\tilde{\gamma}}(X, \dot{X}, t) = \dot{\tau}(t)L_{\alpha,\beta,\gamma}(X, \frac{1}{\dot{\tau}(t)}\dot{X}, \tau(t))$$

As a result the entire family of accelerated functions in reality represents as a single curve, travelling at different speeds.

One of the main claims in [14] was that satisfying the Euler-Lagrange equation implied that Nesterov's algorithm minimized the action functional over the $X \in \mathcal{C}([t_1, t_2], \mathbb{R}^d)$. Interestingly, this claim seems to be untrue in general case, according to [15], and some further restrictions are required to minimize the action potential, by considering the second variation of the Lagrangian.

In fact, satisfying the Euler - Lagrange equation only makes the action stationary, which is proved later.

5.4 Action Functional for Bregman Lagrangian

For the Bregman Lagrangian $L_{\alpha,\beta,\gamma}(X, \dot{X}, t)$ one is interested in the difference in the functional, so using Taylor expansion as follows:

$$\begin{aligned}\Delta J[X,h] &= J[X+h] - J[X] \\ &= J[X] + h\dot{J}[X] + \frac{1}{2}h^2\ddot{J}[X] - J[X] + O(h^3) \\ &= \int_{t_1}^{t_2} \frac{d}{dx} hL(X,\dot{X},t)\,dt + \frac{1}{2}\int_{t_1}^{t_2} \frac{d^2}{dx^2} h^2 L(X,\dot{X},t)\,dt + O(h^3).\end{aligned}$$

By implementing $L_{x_1x_2} = \frac{\partial L}{\partial x_1 \partial x_2}$ notation and by neglecting the $O(h^3)$ terms, one gets:

$$\int_{t_1}^{t_2} L_X h + L_{\dot{X}}\dot{h}\,dt + \frac{1}{2}\int_{t_1}^{t_2} L_{XX}h^2 + 2L_{X\dot{X}}\dot{h}h + L_{X\dot{X}}\dot{h}^2\,dt.$$

The first variation is $\phi_1 = \int_{t_1}^{t_2} L_X h + L_{\dot{X}}\dot{h}\,dx$. For simplicity of notation we can rewrite the second variation:

$$\begin{aligned}\phi_2 &= \frac{1}{2}\int_{t_1}^{t_2} L_{XX}h^2 + 2L_{X\dot{X}}\dot{h}h + L_{X\dot{X}}\dot{h}^2\,dt \\ &= \frac{1}{2}\int_{t_1}^{t_2} \left(L_{XX} - \frac{d}{dt}L_{X\dot{X}}\right)h^2 + L_{\dot{X}\dot{X}}\dot{h}^2\,dt \\ &= \frac{1}{2}\int_{t_1}^{t_2} \left(Qh^2 + P\dot{h}^2\right)\,dt,\end{aligned}$$

where $Q = L_{XX} - \frac{d}{dt}L_{X\dot{X}}$ and $P = L_{\dot{X}\dot{X}}\dot{h}^2$.

Definition 5.1. *A point $t \in (t_1, t_2)$ is **conjugate** to t_1 if the **Jacobi's equation** $\frac{d}{dt}(P\dot{h}) - Qh = 0$ admits a solution that vanishes at both t and t_1 but is not trivial (identically zero).*

Thus, as stated in [14], a necessary and sufficient condition for $X \in \mathcal{C}([t_1, t_2], \mathbb{R}^d)$ to minimize J, is that (1) it satisfies the Euler Lagrange equation,(2) there are no conjugate points to t_1 in (t_1, t_2) and that (3) P is positive semi-definite.

5.5 Analysis for vanishing damping

In accordance with [15],one can use Theorem 2 to analyze the effectiveness of NGD in optimizing the action. In this case focusing on a simplest one - dimensional case, where the objective function is quadratic is effective.

Consider $f(x) = \frac{\beta}{2}x^2$. In this case $Q = -\beta t^3$ and $P = t^3$. Immediately one gets that the Jacobi's equation expand as follows:

$$\frac{d}{dt}(t^3\dot{h}) - \beta t^3 h = 0 \implies \ddot{h} + \frac{3}{t}\dot{h} + \beta h = 0.$$

To satisfy the condition (2) in Theorem 2, consider the solutions for which $h(t_1) = 0$. The equations of this form are solved by using Bessel functions[4]. In particular any solution can be written as:

$$h(t) = C\frac{\mathcal{Y}_1(\sqrt{\beta}t)}{t} - C\frac{\mathcal{Y}_1(\sqrt{\beta}t_1)\mathcal{J}_1(\sqrt{\beta}t)}{\mathcal{J}_1(\sqrt{\beta}t)}, \tag{9}$$

where $C > 0$ is constant specified by the initial velocity, and $\mathcal{J}_\alpha$ is the Bessel function of the first kind and $\mathcal{Y}_\alpha$ is the Bessel function of the second kind, where we have that:

$$\mathcal{J}_\alpha(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{n!\Gamma(n+\alpha+1)} \left(\frac{x}{2}\right)^{2n+\alpha}, \quad \mathcal{Y}_\alpha(x) = \frac{\mathcal{J}_\alpha(x)\cos(\alpha\pi) - \mathcal{J}_{-\alpha}(x)}{\sin(\alpha\pi)}, \tag{10}$$

where Γ is the gamma function. If we consider points $t \in (t_1, t_2)$, where $h(t) = o$ the following identity holds:

$$\frac{\mathcal{Y}_1(\sqrt{\beta}t)}{\mathcal{Y}_1(\sqrt{\beta}t_1)} = \frac{\mathcal{J}_1(\sqrt{\beta}t)}{\mathcal{J}_1(\sqrt{\beta}t_1)} \implies \mathcal{Y}_1(\sqrt{\beta}t) = \frac{\mathcal{Y}_1(\sqrt{\beta}t_1)}{\mathcal{J}_1(\sqrt{\beta}t_1)}\mathcal{J}_1(\sqrt{\beta}t), \tag{11}$$

where we can set $K_{\beta,t_1} = \frac{\mathcal{Y}_1(\sqrt{\beta}t_1)}{\mathcal{J}_1(\sqrt{\beta}t_1)}$. We expand according to [6] as follows:

$$\mathcal{J}_1(x) = \sqrt{\frac{2}{\pi x}}\left(\cos\left(x - \frac{3\pi}{4}\right) + O\left(\frac{1}{x}\right)\right), \mathcal{Y}_1(x) = \sqrt{\frac{2}{\pi x}}\left(\sin\left(x - \frac{3\pi}{4}\right) + O\left(\frac{1}{x}\right)\right).$$

Both of which oscillate around 0 with difference in phase. Thus, equation (8) is satisfied for large enough t. I.e. if $|t_2 - t_1|$ is small enough there are no conjugate points in the interval, satisfying condition (2) in Theorem 2, and then Nesterov path minimizes the action over all curves. Otherwise, a Nesterov path is a saddle point for J. Note, however, the optimiality interval $|t_2 - t_1|$ is smaller for larger value of maximum eigenvalue of Hessian of $f(x)$, β. As such, one can discuss what are the conditions on $|t_2 - t_1|$ for a Nesterov path to extremize the action.

Theorem 3. *For β - smooth quadratics, the second variation of the action of Nesterov's Lagrangian on $f(x) = \beta x^2/2$ is an indefinite quadratic form for $|t_2 - t_1| > \sqrt{\frac{40}{\beta}}$, i.e. Nesterov path is a saddle point for J.*

Proof. The second variation of J along $\gamma \in \mathcal{C}([t_1, t_2], \mathbb{R}^d)$ is $\frac{1}{2}\int_{t_1}^{t_2} t^3(\dot{h}(t)^2 - \beta h(t)^2)\, dt$. One can consider a perturbation $\tilde{h}(t)$ modelled by a triangular function on $(c - \epsilon, c + \epsilon)$, where $c \in (t_1, t_2)$ such that for a modification $h(t)$, $\|\tilde{h} - h\|$ is negligible.

$$\tilde{h(t)} = \begin{cases} \dfrac{t - c + \epsilon}{\epsilon}, t \in (c - \epsilon, c) \\ \dfrac{-t + c + \epsilon}{\epsilon}, t \in (c, c + \epsilon). \end{cases}$$

Then for any scaling factor σ, the second variaiton of the action of $\sigma h(t)$ is

$$\phi_2(\sigma h(t)) = -\sigma^2 \frac{\left(\frac{3\beta\epsilon^4}{10} + (\beta c^2 - 3)\epsilon^2 - 3c^2\right) c}{3\epsilon}.$$

[4]Bessel functions are solutions to the Bessel Differential Equations. The Bessel function of the first kind is finite at $x = 0$ and the Bessel function of the second kind diverge at 0.

Since $c > 0, \epsilon > 0$, one gets that the numerator has two real roots in ϵ^2, smaller root is always negative[5], thus admitting only one solution $\epsilon_*^2(u,\beta)$. As $\epsilon_*^2(u,\beta)$ is a decreasing in u as $u \to \infty$, and since $\epsilon_*^2(0,\beta) = \frac{10}{\beta}$ and $\lim_{u\to\infty} \epsilon_*^2(u,\beta) = \frac{3}{beta}$ we know that ϕ_2 changes sign when $\epsilon_* \in [\frac{10}{\beta}, \frac{3}{\beta}]$

For big ϵ for which h vanishes at t_1, t_2, the second variation is indefinite, when $|t_2 - t_1| > 2\epsilon_*$. Then by taking $\sigma \to \infty$ we get the desired result that $|t_2 - t_1| > \sqrt{\frac{40}{\beta}}$. □

Thus, it has been shown that Nesterov Acceleration locally optimizes the functional over all curves, however for larger t this property breaks down. Which means that the essence of acceleration cannot be captured by minimization of the action functional, unless the conditions stated in Chapter 5.4 are met.

6 Riemannian Manifolds and Geometric interpretation

It has been now shown how in a continuous, non - Euclidean setting one can achieve a faster convergence rate by considering a specific Lagrangian for the motion of the accelerated algorithm, to get to exponential rate of convergence. In particular, we have considered a path for an arbitrary space. One can, however, extend this notion to manifolds, generalizing the the result achieve in [14] to Riemannian manifolds, with similar properties.

6.1 Introduction to Manifolds

In order to be able to analyze the paths on the manifolds, one has to thoroughly define mathematical structures used in analysis in differential geometry. In this section important notions from Geometry are introduced, while some previously stated definitions are redefined in the setting of manifolds. In particular, one starts by redefining the objective:

$$\min_{x\in M} f(x),$$

where M is a manifold.

Definition 6.1. *A Manifold M is a toplogical space, such that for each point $p \in M$ has an open neighborhood $\mathcal{U}$ which is homeomorphic[6] to an open subset of a Euclidean space.*

Definition 6.2. *Given a smooth manifold $M \subset \mathbb{R}^n$ and a point $p \in M$, a vector $v \in \mathbb{R}^n$ is tangent vector at p iff there exists a smooth curve $\gamma : \mathbb{R} \to M$ with*

$$\gamma(p) = 0, \dot{\gamma}(p) = v,$$

The set of tangent vectors of M at p is called the tangent space of M at p:

$$T_pM = \{v | \gamma(p) = 0, \dot{\gamma}(p) = v\}.$$

[5] the sign of the roots is determined by $(15 - 5(\beta c^2)\sqrt{25(\beta c^2)^2 - 60(\beta c^2) + 225})$ which is negative for $\beta c^2 > 0$

[6] A function $f : X \to Y$ between topological spaces is homeomorphic if it's continuous, bijective, and it's inverse is also continuous

It is important to recall the interpretation that a tangent vector v at point p has to act on some function $f : M \to \mathbb{R}$ by giving the directional derivative at point p of γ defined on the manifold. In other words we define a tangent vector $X_{\gamma,p}$ as:

$$v = X_{\gamma,p}(f) := \frac{\partial}{\partial\lambda}(f \circ \gamma)(\lambda)\Big|_{\lambda=p},$$

for some $\lambda \in \mathbb{R}$.

Importantly, unlike manifolds, which are topological spaces, which limit our ability to conduct analysis, tangent space is a vector space, allowing us to impose mathematical structures on it, thus proving useful in analysis.

Definition 6.3. *Let V be a vector space, then the dual V^* of V is the set of linear map of linear maps ω:*

$$V^* = \{\omega : V \to \mathbb{R} : \ \omega \ linear\}.$$

Definition 6.4. *Given a smoooth manifold M, for each point $p \in M$; we define the cotangent space at p, denoted by T_p^*M; to be the dual space to T_pM:*

$$(T_p^*M) = (T_pM)^*.$$

Elements of the cotangent space are called the covectors.

Definition 6.5. *Given a manifold M, then tangent bundle of M is the disjoint union of tangent spaces and cotangent bundle of M is the disjoint union of cotangent spaces:*

$$TM = \{(p, v)|p \in M, v \in T_pM\} \quad T^*M = \{(p, v)|p \in M, v \in T_p^*M\}.$$

Definition 6.6. *Riemannian metric, is inner product defined as a map $g_p : T_pM \times T_pM \to \mathbb{R}$ that is smoothly varying and positive semi definite at each point. A pair (M, g), where g is a Riemannian metric and M is a smooth manifold, is called a Riemannian Manifold.*

Intrinsically, given that $g_p : T_pM \times T_pM \to \mathbb{R}$, a Riemannian metric is a (0,2) tensor, some combination of dual vectors, and thus forms a tensor field, elements of which can be expressed through the elements of the dual basis as $g = \sum_{i,j} g_{ij}\,\mathrm{d}x^i \otimes \mathrm{d}x^j.$, where g_{ij} are constants and $\otimes$ is a tensor product.

Using the notion of Riemannian Manifold, one can define a Riemannian gradient, that acts analagously to the gradient used before, and can be used to identify the dierction of the steepest increase of the function defined on a manifold, which has clear implementation in analyzing optimization methods.

Definition 6.7. *The Riemannian gradient denoted* $\mathrm{grad} f(p) \in T_pM$ *at point $p \in M$ of a smooth function $f : M \to \mathbb{R}$ is a tangent vector at p such that*

$$\langle \mathrm{grad} f(p), v\rangle = df(p)v \quad \forall v \in T_pM,$$

where df is the differential of f.[7]

[7]For $v \in T_pM$, the action of the differential df_p on v, is defined by the directional derivative of f at p in the direction of v:$df_p(v) = \lim_{h\to 0} \frac{f(\gamma(h))-f(p)}{h}$, and is an element of the cotangent space.

Put simply, a gradient of a function f from manifold to the reals, is a tangent vector at a point p, that represents the direction in which f increases the fastest.

Definition 6.8. *A vector field on a Riemannian manifold M is a smooth function $X : M \to TM$ such that for each $p \in M$, $X(p) \in T_pM$. The set of all vector fields on M is denoted $\mathcal{X}(M)$*

Definition 6.9. *A geodesic on a smooth manifold M is a curve $\gamma : I \to M$, where $I \subset \mathbb{R}$ which is of minimal local length.*

Intrinsically the notion of a geodesic generalizes the notion of a straight line in Euclidean space onto manifolds. One important property to note is that the geodesic has 0 tangential acceleration, that what makes it the "straightest path". In local coordinates $(x^1, ..., x^n)$, and $\gamma(t) = (x(t)^1, ..., x(t)^n)$ the geodesic equation looks like:[8]

$$\frac{d^2x^i}{dt^2} + \Gamma^i_{jk}\frac{dx^j}{dt}\frac{dx^k}{dt} = 0.$$

Which is equivalent to making the tangential component of the acceleration vector 0.

Equipped with the notion of a geodesic, one requires a way to update and navigate points efficiently along geodesics, paths of minimal distance, to solve optimization problems. Such map is called the exponential map, that enables the smooth transition between points and tangent vectors on the manifold, ensuring that optimization algorithms stay within the manifold's underlying structure. Formally we define an exponential map as follows:

Definition 6.10. *An exponential map $Exp_p : T_pM \to M$ at p is defined by:*

$$Exp_p(v) = \gamma_v(1),$$

where $\gamma : [0, 1] \to \mathbb{R}$ is a unique geodesic in M, with $\gamma_v(0) = p$ and $\dot{\gamma}_v(0) = v$ for any $v \in T_pM$.

In other words, applying the exponential map at point p on the tangent vector v, gives you a point on the manifold, $Exp_p(v)$, which is interpreted as the endpoint of the geodesic, $\gamma_v(1)$. By scaling the tangent vector and applying the exponential map, you obtain a continuous curve on the manifold that represents the geodesic between the initial point p and the endpoint $Exp_p(v)$.

One important property of the exponential map is that it is a diffeomorphism[9] in some neighbourhood inside tangent space at p containing 0. One can define a Logarithmic map, $Log_q : M \to T_pM$, which is defined as the inverse of the Exponential map and takes two points on the manifold and returns the tangent vector at one point that points to the other.

In a way, the logarithmic map facilitates the translation of optimization problems from the manifold space to the tangent space, enabling the application of traditional optimization algorithms, also helping in characterizing convexity properties, effectively providing a tool to express geodesically convex functions in the tangent space, which we will get to later.

[8] The Christoffel symbol is $\Gamma^k_{ij} = \frac{1}{2}g^{kl}\left(\frac{\partial g_{il}}{\partial x^j} + \frac{\partial g_{jl}}{\partial x^i} - \frac{\partial g_{ij}}{\partial x^l}\right)$.

[9] A diffeomorphism is a differentiable homeomorphism.

Definition 6.11. *Given points p, $\tilde{p} \in M$, $v \in T_pM$ can be transported to $T_{\tilde{p}}M$ along geodesic γ by a parallel transport function: $\Gamma(\gamma)_p^{\tilde{p}} : T_pM \to T_{\tilde{p}}M$.*

An important property of $\Gamma(\gamma)_p^{\tilde{p}}$ is that it preserves the inner product, i.e. $\forall u, v, \in T_pM, g_p(u, v) = g_{\tilde{p}}(\Gamma(\gamma)_p^{\tilde{p}}(u), \Gamma(\gamma)_p^{\tilde{p}}(v))$. Parallelism helps us compare vectors in curved space, in a way that takes into account the curvature, as it shows how the tangent vector changes as it transported from point to point.

Finally to differentiate vector fields in a manner that respects the curvature of the underlying space, one needs a notion of covariant derivative of one vector field with respect to another, an analog of partial derivative, but in curved spaces. Incorporating the parallel transport function into the definition of covariant derivative moves the vector field, and only then compares relative vectors.

Definition 6.12. *Given $X, Y \in \mathcal{X}(M)$, the covariant derivative $\nabla_X Y \in \mathcal{X}(M)$ of Y along X is*

$$\nabla_X Y(p) = \lim_{h \to 0} \frac{\Gamma^p_{\gamma(h)}(\gamma) Y(\gamma(h)) - Y(p)}{h},$$

where gamma is an integral curve of X such that $\gamma(0) = p$

6.2 Convexity in Riemannian Manifolds

After stating the defining notions connected to manifolds, it is possible to analyze the optimization algorithms of functions defined on the manifold. Similar to the GD and NGD in Euclidian space, the convergence rate of various algorithms would be different depending on the convexity of the objective function. However, the notion of convexity on manifolds differs from that of regular functions in Eulcidian space and is categrized into geodesically convex, geodesically $\alpha - strongly - convex$ and geodesically $\lambda - weakly - quasi - convex$ functions.

Definition 6.13. *Let $f : M \to \mathbb{R}$. Then f is:*

1. *geodesically convex if for any $p, \tilde{p} \in M$ and a geodesic γ,*

$$f(\gamma(t)) \leq (1-t)f(p) + tf(\tilde{p}) \quad \forall t \in [0,1]].$$

2. *geodesically $\alpha - strongly - convex$ for some $\alpha > 0$ if*

$$f(p) - f(\tilde{p}) \geq \langle \text{grad} f(\tilde{p}), Log_{\tilde{p}}(p) \rangle + \frac{\alpha}{2} ||Log_{\tilde{p}}(p)||^2.$$

3. *geodesically $\lambda - weakly - quasi - convex$ for some $\lambda \in (0, 1]$ if:*

$$\lambda(f(p) - f(\tilde{p})) \geq \langle \text{grad} f(\tilde{p}), Log_{\tilde{p}}(p) \rangle.$$

Here the fact that for geodesically convex function the inequality $f(p) - f(\tilde{p}) \geq \langle \text{grad} f(\tilde{p}), Log_{\tilde{p}}(p) \rangle$ has been implemented.

Proof. Let $\gamma : [0,1] \to M$ be a geodesic connecting $\tilde{p}$ and p, such that $\gamma(0) = \tilde{p}$ and $\gamma(1) = p$. By convexity of f one immediately gets

$$f(\gamma(t)) \leq (1-t)f(\tilde{p}) + tf(p) \quad t \in [0,1].$$

One can use the exponential map $Exp_p : T_pM \to M$, such that $p = Exp_{\tilde{p}}(tv)$, where $v = Log_{\tilde{p}}p \in T_pM$.

Consider a Taylor expansion of $f(p)$, given by:

$$f(p) \approx f(\tilde{p}) + \langle \text{grad} f(\tilde{p}), v \rangle = f(\tilde{p}) + \langle \text{grad} f(\tilde{p}), Log_{\tilde{p}}p \rangle.$$

Then, taking $t = 1$; the convexity property gives us:

$$f(p) \leq f(\tilde{p}) + \langle \text{grad} f(\tilde{p}), Log_{\tilde{p}}(p) \rangle.$$

□

In the interest of focusing on the objective of the paper, the main focus will be made on a more general family of functions, i.e. $\lambda-weakly-quasi-convex$ functions, hence deriving a more general result. As for the strongly convex functions, the derivation of the convergence rate is described in detail in [5]. Additionally, since geodesically convex function is $\lambda-weakly-quasi-convex$ with $\lambda = 1$, this case also includes convex functions. One can additionally note that A local minimum of a geodesically convex or $\lambda-weakly-quasi-convex$ function is also a global minimum, as it has been stated in [1]. We further define $L-smooth$ functions, and assume that all of the functions f in our investigation are $L-smooth$.

Definition 6.14. *A function $f : M \to \mathbb{R}$ is called L - smooth $\forall p, \tilde{p} \in M$ and a geodesic γ connecting them if:*

$$\|\text{grad} f(p) - \Gamma_{\tilde{p}}^{p}(\gamma)\| \leq L * length(\gamma).$$

6.3 Euler Lagrange equation for the Bregman Lagrangian on a manifold

Throughout the paper ,it is assumed that the solutions to the differential equations remain inside a geodesically uniquely convex subset A of the manifold[10], with the property that $diam(A)$[11] $\leq D \in \mathbb{R}$. Lastly, that the sectional curvature, $K(\sigma)$, that measures the geometric property of the manifolds, by taking perpendicular vectors and see how the angle between them changes as they are parallel transported along the parallelogram, defined by two vectors, u, v is bounded from below by K_{min} on A. Note that for $K(\sigma) > 0$, the space is locally spherical and for $K(\sigma) < 0$ the space is locally hyperbolic.

In this section the Lagrangian satisfying the motion of the algorithm, provided in [5] is used to derive the Euler Lagrange equation of the system and the derrivation of the convergence rate is

[10] any two points in M can be connected by a geodesic

[11] $diam(A) = sup\{d(x,y) : x, y \in A\}$

provided. Consider a path on the manifold M described in coordinates by

$$(x(t), \dot{x}(t)) = (q_1(t), \ldots, q_n(t), v_1(t), \ldots, v_n(t)).$$

Then for For $k = 1, \ldots, n$, Lagrangian for the system is given by

$$\mathcal{L}_{\alpha,\beta,\gamma} = \frac{1}{2} e^{\lambda^{-1}\zeta\gamma_t - \alpha_t} \sum_i \sum_j g_{ij} v^i(t) v^j(t) - e^{\alpha_t + \beta_t + \lambda^{-1}\zeta\gamma_t} f(x(t)),$$

where, α, γ, β as before and

$$\zeta = \begin{cases} \sqrt{-K_{min}} D coth(\sqrt{-K_{min}} D), & K_{min} < 0 \\ 1, & K_{min} \geq 0. \end{cases} \tag{12}$$

Partial derivative with respect to $\dot{q}^k = v^k$ gives:

$$\frac{\partial}{\partial v^k} \mathcal{L}_{\alpha,\beta,\gamma} = \frac{1}{2} e^{\lambda^{-1}\zeta\gamma_t - \alpha_t} \left(\sum_i g_{ik} v^i(t) + \sum_j g_{kj} v^j(t) \right) = e^{\lambda^{-1}\zeta\gamma_t - \alpha(t)} \sum_i g_{ik} v^i(t),$$

where the symmetric property of the metric tensor was used, i.e. $g_{ik} = g_{ki}$. Further, using the fact that $\frac{d}{dt}\left(\sum_i g_{ij} v^i(t)\right) = \sum_i \sum_j \frac{\partial g_{ij}}{\partial q_j} \frac{\partial q_j}{\partial t} v^i(t) + \sum_i g_{ik} \frac{dv^i(t)}{dt} = \sum_i \sum_j \frac{\partial g_{ij}}{\partial q_j} v^j(t) v^i(t) + \sum_i g_{ij} \frac{dv^i(t)}{dt}$ we get

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial v_k}\right) = e^{\lambda^{-1}\zeta\gamma_t - \alpha_t} \sum_{i=1}^n g_{ik} \frac{dv_i}{dt}(t) + e^{\lambda^{-1}\zeta\gamma_t - \alpha_t} \sum_{i,j=1}^n \frac{\partial g_{kj}}{\partial q_i} v_i(t) v_j(t) + (\lambda^{-1}\zeta\dot{\gamma}_t - \dot{\alpha}_t) e^{\lambda^{-1}\zeta\gamma_t - \alpha_t} \sum_{i=1}^n g_{ik} v_i(t),$$

$$\frac{\partial \mathcal{L}}{\partial q_k} = \frac{1}{2} e^{\lambda^{-1}\zeta\gamma_t - \alpha_t} \sum_{i,j=1}^n \frac{\partial g_{ij}}{\partial q_k} v_i(t) v_j(t) - e^{\alpha_t + \beta_t + \lambda^{-1}\zeta\gamma_t} \frac{\partial f}{\partial q_k}.$$

Multiplying both terms by $e^{\alpha_t - \lambda^{-1}\zeta\gamma_t}$, the Euler–Lagrange equations for the Bregman Lagrangian $L_{\alpha,\beta,\gamma}$ are given, for $k = 1, \ldots, n$, by

$$\sum_{i=1}^n g_{ik} \frac{dv_i}{dt}(t) + \sum_{i,j=1}^n \frac{\partial g_{kj}}{\partial q_i} v_i(t) v_j(t) + (\lambda^{-1}\zeta\dot{\gamma}_t - \dot{\alpha}_t) \sum_{i=1}^n g_{ik} v_i(t) = \frac{1}{2} \sum_{i,j=1}^n \frac{\partial g_{ij}}{\partial q_k} v_i(t) v_j(t) + e^{2\alpha_t + \beta_t} \frac{\partial f}{\partial q_k}.$$

Rearranging terms, and multiplying by the matrix (g^{ij}) which is the inverse of (g_{ij}), as well as decomposing the metric tensor into symmetric and skew symmetric components, we get, for $k = 1, \ldots n$, the equation

$$\left(\frac{dv_k}{dt}(t) + \sum_{i,j=1}^n \Gamma_{ij}^k(x(t)) v_i(t) v_j(t) \right) + (\lambda^{-1}\zeta\dot{\gamma}_t - \dot{\alpha}_t) v_k(t) + e^{2\alpha t + \beta t} (gradf(x(t)))_k = 0,$$

So the resulting Euler Lagrange equation becomes:

$$\boxed{\nabla_{\dot{X}} \dot{X} + (\lambda^{-1}\zeta\dot{\gamma}t - \dot{\alpha}_t)\dot{X} + e^{2\alpha_t + \beta t} \mathrm{grad} f(X) = 0.} \tag{13}$$

6.4 Convergence of accelerated algorithm on the Riemannian Manifold

Using this Euler Lagrange formulation, one is in the position to derive the convergence rate for the algorithm, i.e. how fast the solution X to the Bregman Lagrangian approaches the minimizer of the function $f : M \to \mathbb{R}$, defined on the manifold.

Similar to the derivation of convergence rate in Euclidean space, the derivation includes the use of Lyapanov functions. Following the proof in [5], one starts by defining the following energy functional:

$$\mathcal{E}(t) = \lambda^2 e^{\beta_t}(f(X) - f(x^*)) + \frac{1}{2}(\zeta - 1)\|Log_X(x^*)\|^2 + \frac{1}{2}\|\lambda e^{-\alpha_t}\dot{X} - Log_X(x^*)\|^2.$$

In order to find the time derrivative of the Lyupanov function we need to use the fact that $\frac{d}{dt}||Log_{X(t)}(q)||^2 = 2\langle Log_{X(t)}(q), \nabla_{\dot{X}} Log_{X(t)}(q)\rangle = 2\langle Log_{X(t)}(q), -\dot{X(t)}\rangle$.

Taking the time derivative of the energy functional one gets:

$$\begin{aligned}
\dot{\mathcal{E}}(t) &= \lambda^2 \dot{\beta}_t e^{\beta_t}(f(X) - f(x^*)) + \lambda^2 e^{\beta_t}\langle \text{grad} f(X), \dot{X}\rangle + (\zeta - 1)\langle Log_X(x^*), -\dot{X}\rangle \\
&\quad + \langle \lambda e^{-\alpha_t}\dot{X} - Log_X(x^*), -\dot{\alpha}_t \lambda e^{-\alpha_t}\dot{X} + \lambda e^{-\alpha_t}\nabla_{\dot{X}}\dot{X} - \nabla_{\dot{X}} Log_X(x^*)\rangle \\
&= \lambda^2 \dot{\beta}_t e^{\beta_t}(f(X) - f(x^*)) + \lambda^2 e^{\beta_t}\langle \text{grad} f(X), \dot{X}\rangle + (\zeta - 1)\langle Log_X(x^*), -\dot{X}\rangle \\
&\quad + \langle \lambda e^{-\alpha_t}\dot{X} - Log_X(x^*), \lambda e^{-\alpha_t}(-\dot{\alpha}_t \dot{X} + \nabla_{\dot{X}}\dot{X}) - \nabla_{\dot{X}} Log_X(x^*)\rangle.
\end{aligned}$$

Note, expanding the Bregman Lagrangian gives :

$$-\dot{\alpha}_t \dot{X} + \nabla_{\dot{X}}\dot{X} = \lambda^{-1}\zeta e^{\alpha_t}\dot{X} - e^{2\alpha_t + \beta t}\text{grad} f(X).$$

So substituting back and further expanding the brackets:

$$\begin{aligned}
\dot{\mathcal{E}}(t) &= \lambda^2 \dot{\beta}_t e^{\beta_t}(f(X) - f(x^*)) + \lambda^2 e^{\beta_t}\langle \text{grad} f(X), \dot{X}\rangle + (\zeta - 1)\langle Log_X(x^*), -\dot{X}\rangle \\
&\quad + \langle \lambda e^{-\alpha_t}\dot{X} - Log_X(x^*), \lambda e^{-\alpha_t}(\lambda^{-1}\zeta e^{\alpha_t}\dot{X} - e^{2\alpha_t + \beta t}\text{grad} f(X)) - \nabla_{\dot{X}} Log_X(x^*)\rangle \\
&= \lambda^2 \dot{\beta}_t e^{\beta_t}(f(X) - f(x^*)) + \lambda^2 e^{\beta_t}\langle \text{grad} f(X), \dot{X}\rangle + (\zeta - 1)\langle Log_X(x^*), -\dot{X}\rangle - \lambda\zeta e^{-\alpha_t}\langle \dot{X}, \dot{X}\rangle \\
&\quad - \lambda^2 e^{\beta_t}\langle \dot{X}, \text{grad} f(X)\rangle - \lambda e^{-\alpha_t}\langle \dot{X}, \nabla_{\dot{X}} Log_X(x^*)\rangle + \zeta\langle Log_X(x^*), \dot{X}\rangle \\
&\quad + \lambda e^{\alpha_t + \beta_t}\langle Log_X(x^*), \text{grad} f(X)\rangle + \langle Log_X(x^*), \nabla_{\dot{X}} Log_X(x^*)\rangle.
\end{aligned}$$

Cancelling the $\langle \text{grad} f(X), \dot{X}\rangle$ and $\langle Log_X(x^*), -\dot{X}\rangle$ terms out we get:

$$\begin{aligned}
\dot{\mathcal{E}}(t) &= \lambda^2 \dot{\beta}_t e^{\beta_t}(f(X) - f(x^*)) + \lambda e^{\alpha_t + \beta_t}\langle Log_X(x^*), \text{grad} f(X)\rangle \\
&\quad - \lambda\zeta e^{\alpha_t}\langle \dot{X}, \dot{X}\rangle - \lambda e^{-\alpha_t}\langle \dot{X}, \nabla_{\dot{X}} Log_X(x^*)\rangle \\
&= \lambda e^{\beta_t}[\dot{\beta}_t \lambda(f(X) - f(x^*)) + e^{\alpha_t}\langle Log_X(x^*), \text{grad} f(X)\rangle] \\
&\quad - \lambda e^{-\alpha_t}[\zeta\langle \dot{X}, \dot{X}\rangle + \langle \dot{X}, \nabla_{\dot{X}} Log_X(x^*)\rangle].
\end{aligned}$$

By using the fact, that f is geodesically $\lambda - weakly - quasi - convex$, one gets that:

$$\lambda(f(X) - f(x^*)) + \langle Log_X(x^*), \text{grad} f(X)\rangle \leq 0.$$

Under the ideal scaling condition: $\dot{\beta}_t \leq e^{\alpha_t}$ and using the $\langle \nabla_{\dot{X}} Log_X(p), -\dot{X}\rangle \leq \zeta||\dot{X}||^2$, proved in [5] as well as $\zeta\langle \dot{X}, \dot{X}\rangle + \langle \dot{X}, \nabla_{\dot{X}} Log_X(x^*)\rangle \geq 0$ one gets:

$$\lambda e^{\beta_t}[\dot{\beta}_t \lambda(f(X) - f(x^*)) + e^{\alpha_t}\langle Log_X(x^*), \text{grad} f(X)\rangle] \leq 0$$

$$-\lambda e^{-\alpha_t}[\zeta\langle \dot{X}, \dot{X}\rangle + \langle \dot{X}, \nabla_{\dot{X}} Log_X(x^*)\rangle] \leq 0.$$

Which shows that $\dot{\mathcal{E}}(t) \leq 0$, making it a decreasing function.

$$\lambda^2 e^{\beta_t}(f(X) - f(x^*)) \leq \lambda^2 e^{\beta_t}(f(X) - f(x^*)) + \frac{1}{2}(\zeta - 1)\|Log_X(x^*)\|^2 + \frac{1}{2}\|\lambda e^{-\alpha_t}\dot{X} - Log_X(x^*)\|^2$$
$$= \mathcal{E}(t) \leq \mathcal{E}(0) = \lambda^2 e^{\beta_0}(f(x_0) - f(x^*)) + \frac{1}{2}\zeta\|Log_{x_0}(x^*)\|^2,$$

thus:

$$\boxed{f(X) - f(x^*) \leq \frac{\lambda^2 e^{\beta_0}(f(x_0) - f(x^*)) + \frac{1}{2}\zeta\|Log_{x_0}(x^*)\|^2}{\lambda^2 e^{\beta_t}}.} \tag{14}$$

Giving the desired exponential convergence rate.

As with the non-manifold setting, the discretization of the p-Bregman Lagrangian could be seen in [5] (See Appendix B).

6.5 Time invariance

The time invariance property of the solutions to the Lagrangian is extended analogously to the Manifolds setting. In particular, we get that if the curve $X(t)$ satisfying the equation (13), the the parameterized curve would satisfy $\mathcal{L}_{\tilde{\alpha}}, \tilde{\beta}, \tilde{\gamma}$ with $\tilde{\alpha}_t = \alpha_{\tau(t)} + log\dot{\tau}(t), \tilde{\beta}_t = \beta_{\tau(t)}, \tilde{\gamma}_t = \gamma_{\tau(t)}$. In particular, as before, we get that:

$$L_{\tilde{\alpha},\tilde{\beta},\tilde{\gamma}}(X, \dot{X}, t) = \dot{\tau}(t) L_{\alpha,\beta,\gamma}(X, \frac{1}{\dot{\tau}(t)}\dot{X}, \tau(t)),$$

which again implies that the family of solutions of the Lagrangian simply represent the same carve on the manifold at different speed.

7 Applications and Extensions to the work

Any system that requires minimization or maximization of a convex function presents an optimization problem, that could be solved using the methods developed in Chapters 2 - 5. In particular, methods that include discretezations of p - Bregman Lagrangian will yield higher convergence rates. Additionally, a system that has some underlying manifold topological structure defined on it, or could be reduced to a topological space, has a potential to be solved using geometric methods discussed in the paper, if one can define a Riemannian metric on it, with a weakly convex function defined on it.

It was of general interest to explore real world optimization problems, both in a conventional setting and with manifold structure defined on a system. As such, relying on the framework developed in [7] and [3], in this section the application of optimization are discussed in the context of penalty function approximation problem and Bach Normalization in Deep Learning Networks.

It is worth noting that despite the methods developed under discretizations of the sub-families of Bregman Lagrangian being quite efficient, there are other known optimization techniques that could be used in the manifold optimization problems, like Riemannian Gradient Descent, Stochastic Gradient Descent, Newtons ethods, etc.

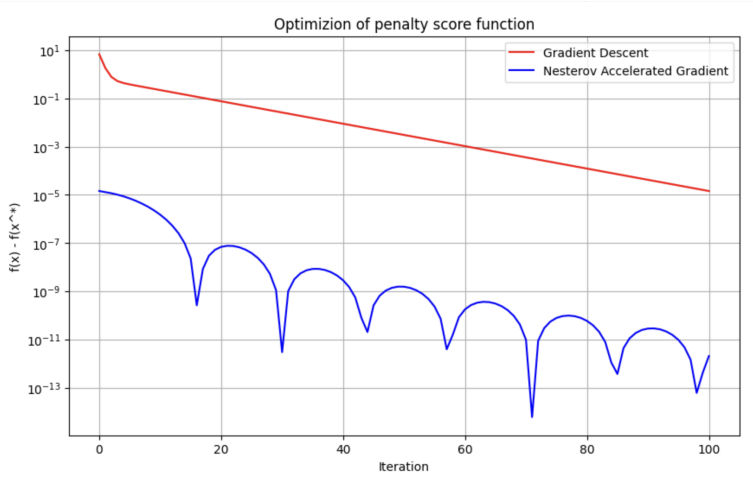


Figure 3: GD vs NGD on PSF for a randomly generated parameters in $\mathbb{R}^2$

7.1 Penalty function approximation problem

Classically, a penalty function approximation problem is a specific case of a norm approximation problem, and serves as a generalization of ℓ_p - norm approximation problem[3]. The problem is stated as follows:

$$\begin{aligned} min \quad & \phi(r_1) + ... + \phi(r_n) \\ subject\ to \quad & r = Ax - b, \end{aligned}$$

where $A \in \mathbb{R}^{mxn}$, $b \in \mathbb{R}^m$, $x \in \mathbb{R}^n$ is the variable.A function $\phi : \mathbb{R} \to \mathbb{R}$, assumed to be convex, is called a penalty function, and it defines the norm for each residual, r, that varies with x.

The problem involves minimisation of the penalty function. Some common examples of penalty functions include log-barrier, quadratic and deadzone linear functions.

We can consider a specific case of penalty function approximation problem, with $\phi(r_i) = r_i^2$, i.e. a quadratic function, and $A \in \mathbb{R}^{2x2}$. Then the problem simplifies to a least-squares problem, namely:

$$\min_{x \in \mathbb{R}^2} \sum_{i=2}^{2} r_i^2 = \min_{x \in \mathbb{R}^2} \|Ax - b\|_2^2.$$

Figure 3 offers a simulation for a randomly generated A, b, with GD and Nesterov (p - Bregman discretezation with $p = 2$).

7.2 Batch Normalisation

Bath Normalization is the technique used for training neural networks more efficiently, by normalising the input at each layer, rather than just the input. The normalization process involves transforming the output values of the previous layer $x = x_{i...N}$ (input values, if it's input layer), such that it has 0 mean and variance of 1. The deeper explanation of this process is discussed in [8].

By defining w to be a vector of scaling parameters associated to x, the Batch Normalization of $z := w^\top x$ can be defined by standard normalization function:

$$BN(z) = \frac{z - \mathbf{E}(z)}{\sqrt{Var(z)}} = \frac{w^\top (x - \mathbf{E}(x))}{\sqrt{w^\top R_{xx} w}},$$

where R_{xx} is the covariance matrix of x. As such, the space of weight vectors, w, can be viewed as a topological space underlying the Riemannian manifold, where each point corresponds to a point on a manifold. Additionally, note, that under normalization of scale parameters, w, we get that for $u = \frac{w}{|w|}$:

$$BN(z) = \frac{u^\top (x - \mathbf{E}(x))}{\sqrt{u^\top R_{xx} u}}.$$

Additionally, by noting that $\frac{\partial (w^\top x)}{\partial x} = \frac{\partial (u^\top x)}{\partial x}$ and $\frac{\partial (z)}{\partial w} = \frac{1}{w}\frac{\partial (z)}{\partial u}$, ensures that the use of Batch Normalization doesn't let the learning rate of the algorithm to explode.

The goal is focus on the family of manifolds where the scaled vector, z collapses to a point. One of the families with this property, presented in [4] was a Grassman Manifold, $\mathcal{G}(p, d)$, for $p = 1$. In particular, $\mathcal{G}(p, d)$ represents a set of p-dimensional subspace of $\mathbb{R}^d$, so the space of scaled vectors, z, forms a 1 - dimensional subspace of $\mathbb{R}^d$.

Since $BN(z)$ is invariant under linear scaling[12], by viewing the scaling vectors, w, as a point on $\mathcal{G}(1, d)$ sphere, and equipping the Grassman manifold with Riemannian metric, so called Grassmanian manifold, the optimization problem can be formulated in terms of optimizing the loss function, L, the form of which depends on the specific context of the problem, and is stated as :

$$\min_{X \in \mathcal{M}} \quad L(X), \quad \mathcal{M} = S^{d_1 - 1} \times \cdots S^{d_m - 1} \times \mathbb{R}^l,$$

where $n_1, ..., n_m$ are the dimensions of weight vectors, m is the number of weight vectors and l is the number of other parameters like biases, offset parameters, etc(not essential part of this problem).

Clearly, the porblem becomes an optimization problem on multiple spheres, and since Riemannian metric is defined on $\mathcal{M}$, one can use the discretezation of p - Bregman Lagrangian methods, to solve the optimization problem, by constraining the Loss function to be $\lambda - weakly - quasi \quad convex$, achieving polynomial convergence rates derived in [5].

8 Conclusion

The exploration of acceleration in optimization, viewed through the lens of geometry and classical physics has revealed a connection between mathematical structures, like Lagrangians/action functionals, and modern optimization techniques. By transitioning from discrete optimization techniques to continuous time limits, one gains a deeper understanding of these methods. The variational approach has proved itself to be a valid method for analyzing existing algorithms and inspiring the development of better accelerated methods, through discretization procedure.

[12] $BN(cz) = \frac{cw^\top (x - \mathbf{E}(x))}{\sqrt{c^2 w^\top R_{xx} w}} = BN(z)$.

The extension of such findings to Riemannian manifolds represents a significant advancement, broadening the scope of application to a wider set of problems, that have topological structures underlying them, that could be represented as a manifold[13]. The practical method of both approaches is demonstrated through their application in problems like penalty function approximation, batch normalization, and of course many more.

The scope of this investigation couldn't have possibly captured all the recent developments in the area, and present just a small part of the research in the field. Some of the interesting cases, like the investigation of convergence of strongly convex function, were left out but are of great interest and significance. Additionally, the Hamiltonian framework for analysis presents another way to look at the problem but was also left out of the investigation. Lastly, unfortunately, some interesting properties of the ODE, like Gauge invariance, were not investigated and numerical results illustrating optimization on Riemannian manifolds were not presented due to the difficulty of implementation, which could have served as a demonstration of the methods analyzed in the paper.

As for future research, Nesterov's accelerated methods have been extended to other settings, like, for instance, the stochastic setting. Naturally, it would be great to analyze these in the Lagrangian framework too. In fact, many other interesting accelerated optimization methods that exist today, like mirror descent or stochastic gradient descent could be studied in a similar manner and could potentially present interesting results.

Lastly, during the investigation, a variety of very interesting applications were considered, for which the objective function was not convex. It would be interesting to conduct similar research in that area, under, of course, different constraints, on methods like Adaptive Moment Estimation, AdaGrad, etc.

[13]Under given conditions.

References

[1] Foivos Alimisis, Antonio Orvieto, Gary Bécigneul, and Aurelien Lucchi. A continuous-time perspective for modeling acceleration in riemannian optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 1297–1307. PMLR, 2020.

[2] Siva Balakrishnan. Lecture 2. *10-725 Convex Optimization*, 2023.

[3] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[4] Minhyung Cho and Jaehyung Lee. Riemannian approach to batch normalization. *Advances in Neural Information Processing Systems*, 30, 2017.

[5] Valentin Duruisseaux and Melvin Leok. A variational formulation of accelerated optimization on riemannian manifolds. *SIAM Journal on Mathematics of Data Science*, 4(2):649–674, 2022.

[6] F.R.S. G.N. Watson, Sc.D. *a treatise on the theory of bessel functions*. 1922.

[7] Jiang Hu, Xin Liu, Zai-Wen Wen, and Ya-Xiang Yuan. A brief introduction to manifold optimization. *Journal of the Operations Research Society of China*, 8:199–248, 2020.

[8] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[9] Daniel Liberzon. *Calculus of variations and Optimal Control Theory: A concise introduction*. World Publishing Corporation, 2013.

[10] Y Nesterov. A method of solving a convex programming problem with convergence rate mathcal {O}(1/kˆ {2}). In *Sov. Math. Dokl*, volume 27.

[11] B.T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[12] Weijie Su, Stephen Boyd, and Emmanuel Candes. A differential equation for modeling nesterov's accelerated gradient method: theory and insights. *Advances in neural information processing systems*, 27, 2014.

[13] Adrien Taylor, Bryan Van Scoy, and Laurent Lessard. Lyapunov functions for first-order methods: Tight automated convergence guarantees. In *International Conference on Machine Learning*, pages 4897–4906. PMLR, 2018.

[14] Andre Wibisono, Ashia C Wilson, and Michael I Jordan. A variational perspective on accelerated methods in optimization. *proceedings of the National Academy of Sciences*, 113(47):E7351–E7358, 2016.

[15] Peiyuan Zhang, Antonio Orvieto, and Hadi Daneshmand. Rethinking the variational interpretation of accelerated optimization methods. *Advances in Neural Information Processing Systems*, 34:14396–14406, 2021.

9 Appendicies

9.1 Appendix A: Numerical Results

All numerical results are presented in Python, as it;s easier to implement. The functions for the experiments were highly inspired by [**?**], yet were slightly modified in order to obtain some original results. In particular 2 functions were analyzed when comparing ODE to NGD.

Gradient Descent Algorithm

```
def gradient_descent(f, grad_f, x0, s=0.01, n_iters=100):
    x = x0
    fs = []
    for k in range(n_iters):
        x -= s * grad_f(x)
        fs.append(f(x))
    return x, fs
```

Nesterov Accelerated Gradient Descent Algorithm

```
# Nesterov's Accelerated Gradient Algorithm
def nesterov_accelerated_gradient(f, grad_f, x0, s=0.01, n_iters=100):
    x, y = x0, x0
    fs = []
    for k in range(1, n_iters + 1):
        grad = grad_f(y)
        x_k1 = y - s * grad
        y = x_k1 + (k - 1) / (k + 2) * (x_k1 - x)
        x = x_k1
        fs.append(f(x))
    return x, fs
```

The following optimization parameters were chose for the Figure 1. In particular, for both of the simulations the chosen number of iterations was $n = 100$ iterations, with the initial position starting at $x_0 = 10$, in order to keep the simulations under the same condition. However, the chosen step size is different, to highlight the features of the simulations. In particular, in (a) of Figure 1, a step size was different order of magnitudes tried, until the quadratic and linear relation is visible for different algorithms. On(b), the focus was on visualizing the oscillatory nature of the Nesterov ODE.

```
'''
Graph A:  initial x = 10; step size s= 0.0001
Graph B: x0 = [10 ,10], s = 0.1
'''
```

To implement Nesterov ODE, I used the SciPy library to solve the initial value problem that presented the ODE itself. On the graph, I chose different step sizes, of different orders of magnitude, to visualize how NGD approaches the ODE trajectory under $t \to \infty$. In the code, v is the velocity, or $\dot{X}$ and a is the acceleration, or $\ddot{X}$.

Nesterov ODE simulation

```
# ODE for NGD
def ode_system(t, X):
    x, v = X[:2], X[2:]
    a = -3/t * v - grad_f(x)
    return np.hstack((v, a))

# Solve the ODE
ts = [0.01, iter * np.sqrt(s)]  # time != 0 to not divide by 0
initial_conditions = np.hstack((x0, [0, 0]))
sol = solve_ivp(ode_system, ts, initial_conditions,
    t_eval=np.linspace(ts[0], ts[1], iter))
```

In order to implement the penalty function, the same procedure as for Figure 1 was used. However, the objective was different, and included randomly generating matrix A, and vector b, as well as using .dot() for matrix multiplication.

Penalty Loss function parameters for simulation

```
    #Ranomly generated A,b
    A = np.random.randn(2, 2)
    b = np.random.randn(2)
    # Penalty loss funtion
    f = lambda x: np.sum((A.dot(x) - b)**2)
    grad_f = lambda x: 2 * A.T.dot(A.dot(x) - b)
```

9.2 Appendix B: Discretizations of algorithms

9.2.1 Rate matching discretization of Nesterov ODE

As presented in [14], a rate matching discretization is:

$$\begin{aligned} x_{k+1} &= \frac{p}{k+p} z_k + \frac{k}{k+p} y_k \\ z_k &= \arg\min_z \left\{ Cpk^{(p-1)} \langle \nabla f(y_k), z \rangle + \frac{1}{\epsilon} D_h(z, z_{k-1}) \right\}, \end{aligned} \tag{15}$$

where $k^{(p-1)} := k(k+1)\cdots(k+p-2)$.

Under certain conditions, the specific case of this general fomrulation is NGD.

9.2.2 Discretization of ODE on manifolds

The discretisation of the manifold is presented in [1] as:

$$\begin{aligned} X_{k+1} &= \mathrm{Exp}_{X_k}(ha_k), \\ V_{k+1} &= \Gamma_{X_k}^{X_{k+1}} a_k, \end{aligned} \tag{16}$$

where $b_k = 1 - \frac{\zeta p + \lambda}{\lambda k}$, and under different Versions of the algorithm, in particular derivation through semi-implicit Euler scheme(I) or Nesterov's derivation (II), we get:

$$\begin{aligned} (I) &: a_k = b_k V_k - h c_k gradf(X_k) \\ (II) &: a_k = b_k V_k - h c_k gradf\left(\mathrm{Exp}_{X_k}(h b_k V_k)\right), \end{aligned}$$

with $c_k = Cp^2(kh)^{p-2}$ and $C, h, p > 0, X_0 \in \mathcal{Q}$ and $V_0 \in T_{X_0}M$