FIRST-ORDER OPTIMIZATION INSPIRED FROM FINITE-TIME CONVERGENT FLOWS

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we investigate the performance of two first-order optimization algorithms, obtained from forward Euler discretization of finite-time optimization flows. These flows are the rescaled-gradient flow (RGF) and the signed-gradient flow (SGF), and consist of non-Lipscthiz or discontinuous dynamical systems that converge locally in finite time to the minima of gradient-dominated functions. We propose an Euler discretization for these first-order finite-time flows, and provide convergence guarantees, in the deterministic and the stochastic setting. We then apply the proposed algorithms to academic examples, as well as deep neural networks training, where we empirically test their performances on the SVHN dataset. Our results show that our schemes demonstrate faster convergences against standard optimization alternatives.

1 INTRODUCTION

Consider the unconstrained minimization problem for a given cost function $f : \mathbb{R}^n \to \mathbb{R}$. When f is sufficiently regular, the standard algorithm in continuous time (dynamical system) is given by

$$\dot{x} = F_{\mathrm{GF}}(x) \triangleq -\nabla f(x) \tag{1}$$

with $\dot{x} \triangleq \frac{\mathrm{d}}{\mathrm{d}t}x(t)$, known as the *gradient flow* (GF). Generalizing GF, the q-*rescaled* GF (q-RGF) Wibisono et al. (2016) given by

$$\dot{x} = F_{q-\mathrm{RGF}}(x) = -c\frac{\nabla f(x)}{\|\nabla f(x)\|_2^{\frac{q-2}{q-1}}} \tag{2}$$

with $c > 0$ and $q \in (1, \infty]$ has an asymptotic convergence rate $f(x(t)) - f(x^\star) = \mathcal{O}\left(\frac{1}{t^{q-1}}\right)$ under mild regularity, for $\|x(0) - x^\star\| > 0$ small enough, where $x^\star \in \mathbb{R}^n$ denotes a local minimizer of f. However, it has been recently proven Romero & Benosman (2020) that q-RGF, as well as q-*signed* GF (q-SGF)

$$\dot{x} = F_{q-\mathrm{SGF}}(x) = -c\,\|\nabla f(x)\|_1^{\frac{1}{q-1}}\,\mathrm{sign}(\nabla f(x)), \tag{3}$$

where $\mathrm{sign}(\cdot)$ denotes the sign function (element-wise), are both finite-time convergent (in continuous-time), provided that f is gradient dominated of order $p \in (1, q)$. In particular, if f is strongly convex, then q-RGF and q-SGF is finite-time convergent for any $q \in (2, \infty]$, since f must be gradient dominated of order $p = 2$. Concerning that many algorithms are inspired from these continuous flows with convergence guarantee, a natural question arises: *what is the convergence rate of the corresponding discrete-time algorithms, which are induced by discretization of continuous-time flows?*

1.1 CONTRIBUTION

In this paper, we investigate the convergence behavior of an Euler discretization for the q-RGF (equation 2) and q-SGF (equation 3). We provide convergence guarantees in terms of closeness of solutions, using results from hybrid dynamical control theory. Furthermore, we provide iteration / sample complexity upper bounds for both the general (deterministic) and stochastic settings. We then test the performance of the proposed algorithms on both synthetic and real-world data in the context of deep learning, namely, on the well-known SVHN dataset.

1.2 Related Work

Propelled by the work of Wang & Elia (2011) and Su et al. (2014), there has been a recent and significant research effort dedicated to analyzing optimization algorithms from the perspective of dynamical systems and control theory, especially in continuous time (Wibisono et al., 2016; Wilson, 2018; Lessard et al., 2016; Fazlyab et al., 2017b; Scieur et al., 2017; Franca et al., 2018; Fazlyab et al., 2018; Fazlyab et al., 2018; Taylor et al., 2018; França et al., 2019a; Orvieto & Lucchi, 2019; Romero et al., 2019; Muehlebach & Jordan, 2019). A major focus within this initiative is in *accceleration*, both in terms of trying to gain new insight into more traditional optimization algorithms from this perspective, or even to exploit the interplay between continuous-time systems and their potential discretizations for novel algorithm design (Muehlebach & Jordan, 2019; Fazlyab et al., 2017a; Shi et al., 2018; Zhang et al., 2018; França et al., 2019b; Wilson et al., 2019). Many of these papers also focus on explicit mappings and matchings of convergence rates from the continuous-time domain into the discrete-time domain.

Connecting ordinary differential equations (ODEs) and their numerical analysis, with optimization algorithms is a very important topic, which can be dated back to 1970s, see Botsaris (1978a;b); Zghier (1981); Snyman (1982; 1983); Brockett (1988); Brown (1989). In Helmke & Moore (1994), the authors studied relationships between linear programming, ODEs, and general matrix theory. Further, Schropp (1995) and Schropp & Singer (2000) explored several aspects linking nonlinear dynamical systems to gradient-based optimization, including nonlinear constraints.

Tools from Lyapunov stability theory are often employed for the analysis, mainly because there already exists a rich body of works within the nonlinear systems and control theory community for this purpose. In particular, typically in previous works, one seeks asymptotically Lyapunov stable gradient-based systems with an equilibrium (stationary point) at an isolated extremum of the given cost function, thus certifying local convergence. Naturally, the global asymptotic stability leads to global convergence, though such analysis will typically require the cost function to be strongly convex everywhere.

For physical systems, a Lyapunov function can often be constructed from first principles via some physically meaningful measure of energy (*e.g.*, total energy = potential energy + kinetic energy). In optimization, the situation is somewhat similar in the sense that a suitable Lyapunov function may often be constructed by taking simple surrogates of the objective function as candidates. For instance, $V(x) \triangleq f(x) - f(x^\star)$ can be a good initial candidate. Further, if f is continuously differentiable and $x^\star$ is an isolated stationary point, then another alternative is $V(x) \triangleq \|\nabla f(x)\|^2$.

However, most fundamental and applied research conducted in systems and control regarding Lyapunov stability theory deals exclusively with *continuous-time* systems. Unfortunately, (dynamical) stability properties are generally not preserved for simple forward-Euler and sample-and-hold discretizations and control laws (Stuart & Humphries, 1998). Furthermore, practical implementations of optimization algorithms in modern digital computers demand discrete-time. Nonetheless, it has been extensively noted that a vast amount of general Lyapunov-based results (perhaps most) appear to have a discrete-time equivalent.

2 Preliminaries

2.1 Optimization Algorithms as Discrete-Time Systems

Generalizing (1), (2), and (3), consider a continuous-time algorithm (dynamical system) modeled via an ordinary differential equation (ODE)

$$\dot{x} = F(x) \tag{4}$$

for $t \geq 0$, or, more generally, a differential inclusion

$$\dot{x}(t) \in \mathcal{F}(x(t)) \tag{5}$$

a.e. $t \geq 0$ (*e.g.* for the $q = \infty$ case), such that $x(t) \to x^\star$ as $t \to t^\star$. In the case of the q-RGF (2) and q-SGF (3) for f gradient dominated of order $p \in (1, q)$, we have finite-time convergence, and thus $t^\star = t^\star(x(0)) < \infty$.

Most of the popular numerical optimization schemes can be written in a state-space form (*i.e.*, recursively), as

$$X_{k+1} = F_{\mathrm{d}}(k, X_k) \tag{6a}$$
$$x_k = G(X_k) \tag{6b}$$

for $k \in \mathbb{Z}_+ \triangleq \{0, 1, 2, \ldots\}$ and a given $X_0 \in \mathbb{R}^m$ (typically $m \geq n$), where $F_{\mathrm{d}} : \mathbb{Z}_+ \times \mathbb{R}^m \to \mathbb{R}^n$ and $G : \mathbb{R}^m \to \mathbb{R}^n$.

Naturally, (6) can be seen as a discrete-time dynamical system constructed by discretizing (4) in time. In particular, we have $x_k \approx x(t_k)$, where $\{0 = t_0 < t_1 < t_2 < \ldots\}$ denotes a time partition and $x(\cdot)$ a solution to (4) or (5) as appropriate. Therefore, we call X_k and x_k, respectively, the *state* and *output* at time step k. Whenever $F_{\mathrm{d}}(k, X)$ does not depend on k, we will drop the k and thus write $F_{\mathrm{d}}(X)$ instead. Whenever $m = n$, we will denote take $G(X) \triangleq X$ and replace X and X_k by x and x_k, respectively.

Example 1. The standard gradient descent (GD) algorithm

$$x_{k+1} = x_k - \eta \nabla f(x_k) \tag{7}$$

with step size (learning rate) $\eta > 0$ can be readily written in the form (6) by taking $m = n$, $F_{\mathrm{d}}(x) \triangleq x - \eta \nabla f(x)$, and $G(x) \triangleq x$.

- If the step sizes are adaptive, *i.e.* if we replace η by a sequence $\{\eta_k\}$ with $\eta_k > 0$, then we only need to replace $F_{\mathrm{d}}(k, x) \triangleq x - \eta_k \nabla f(x)$, provided that $\{\eta_k\}$ is not computed using feedback from $\{x_k\}$ (*e.g.* through a line search method).
- If we do wish to use *feedback*[1] (and no memory past the most recent output and step size), then we can set $m = n + 1$, $G([x; \eta]) \triangleq x$, and $F_{\mathrm{d}}([x; \eta]) \triangleq [F_{\mathrm{d}}^{(1)}([x; \eta]); F_{\mathrm{d}}^{(2)}([x; \eta])]$, where $F_{\mathrm{d}}^{(1)}([x; \eta]) \triangleq x - \eta \nabla f(x)$, and $F_{\mathrm{d}}^{(2)}$ is a user-defined function that dictates the updates in the step size. In particular, an open-loop (no feedback) adaptive step size $\{\eta_k\}$ may also be achieved under this scenario, provided that it is possible to write $\eta_{k+1} = F_{\mathrm{d}}^{(2)}(\eta_k)$. If this is not possible (and still open-loop step size), then we may take $F_{\mathrm{d}}^{(2)}(k, X) \triangleq \eta_{k+1}$, and of course add a k-argument in F_{d}.
- If we wish to use individual step sizes for each the n components of $\{x_k\}$, then it suffices to take η_k as an n-dimensional vector (thus $m = 2n$), and make appropriate changes in F_{d} and G.

In each of these cases, GD can be seen as a forward-Euler discretization of the GF (1), *i.e.*,

$$x_{k+1} = x_k + \Delta t_k F_{\mathrm{GF}}(x_k) \tag{8}$$

with $F_{\mathrm{GF}} = -\nabla f$ and adaptive time step $\Delta t_k \triangleq t_{k+1} - t_k$ chosen as $\Delta t_k = \eta_k$.

Example 2. The proximal point algorithm (PPA)

$$x_{k+1} = \operatorname*{arg\,min}_{x \in \mathbb{R}^n} \left\{ f(x) + \frac{1}{2\eta_k} \|x - x_k\|_2^2 \right\} \tag{9}$$

with step size $\eta_k > 0$ (open loop, for simplicity) can also be written in the form (6), by taking $m = n$, $F_{\mathrm{d}}(k, x) \triangleq \arg\min_{x' \in \mathbb{R}^n} \{f(x') + \frac{1}{2\eta_k} \|x' - x\|_2^2\}$, and $G(x) \triangleq x$. Naturally, we need to assume sufficient regularity for $F_{\mathrm{d}}(k, x)$ to exist and we must design a consistent way to choose $F_{\mathrm{d}}(k, x)$ when multiple minimizers exist in the definition of $F_{\mathrm{d}}(k, x)$. Alternatively, these conditions must be satisfied, at the very least, at every $(k, x) \in \{(0, x_0), (1, x_1), (2, x_2), \ldots\}$ for a particular chosen initial $x_0 \in \mathbb{R}^n$.

By assuming sufficient regularity, we have $\nabla_x \{f(x) + \frac{1}{2\eta_k} \|x - x_k\|_2^2\}|_{x = x_{k+1}} = 0$, and thus

$$\nabla f(x_{k+1}) + \frac{1}{\eta_k}(x_{k+1} - x_k) = 0 \iff x_{k+1} = x_k + \Delta t_k F_{\mathrm{GF}}(x_{k+1}) \tag{10}$$

with $\Delta t_k = \eta_k$, which is precisely the backward-Euler discretization of the GF (1).

[1] Also known as *closed-loop* design in control-theoretic and reinforcement learning terminology, meaning that $\eta_k = \varphi(k, x_k)$ for some $\varphi : \mathbb{Z}_+ \times \mathbb{R}^n \to \mathbb{R}_+$ that does not depend on $\{X_0, X_1, X_2, \ldots\}$. On the other hand, *open-loop* design can be seen as closed loop with $\varphi(k, \cdot)$ constant for each $k \in \mathbb{Z}_+$.

2.2 CONTINUOUS-TIME CONVERGENCE OF q-RGF AND q-SGF

In this section, we review the necessary conditions to ensure finite-time convergence of these flows. Here the hyperparameter $c > 0$ in equation 2 and equation 3 will not be explicitly denoted in $F_{q-\mathrm{RGF}}, F_{q-\mathrm{SGF}}$. Next, borrowing terminologies from Wilson et al. (2019), we define the notion of gradient dominance with order p as following.

Assumption 1 (Gradient Dominance of Order p)**.** For a continuously differentiable function f, we assume the function f is *μ-gradient dominated of order* $p \in (1, \infty]$ ($\mu > 0$), i.e.,

$$\frac{p-1}{p}\|\nabla f(x)\|_2^{\frac{p}{p-1}} \geq \mu^{\frac{1}{p-1}}(f(x) - f(x^\star)), \quad \forall\, x \in \mathbb{R}^n, \tag{11}$$

where $x^\star \in \arg\min_{x\in\mathbb{R}^n} f(x)$ is the local minimizer, also we denote the optimal value $f^\star \triangleq f(x^\star)$.

Remark 1. It can be proved that continuously differentiable strongly convex functions are gradient dominated of order $p = 2$. In fact, gradient dominance is usually defined exclusively for order $p = 2$, often referred to as the Polyak-Łojasiewicz (PL) inequality, which was introduced by Polyak (1963) to relax the (strong) convexity assumption commonly used to show convergence of the GD algorithm (7). The PL inequality can also be used to relax convexity assumptions of similar gradient and proximal-gradient methods (Karimi et al., 2016; Attouch & Bolte, 2009). Furthermore, if f is gradient dominated (of any order) w.r.t. $x^\star$, then $x^\star$ is an isolated stationary point of f.

Our adopted generalized notion of gradient dominance is strongly tied to the Łojasiewicz gradient inequality from real analytic geometry, established by Łojasiewicz (1963; 1965)[2] independently and simultaneously from Polyak (1963), and generalizing the PL inequality. More precisely, this inequality is typically written as: for some $C > 0$ and $\theta \in \left(\frac{1}{2}, 1\right]$,

$$\|\nabla f(x)\|_2 \geq C \cdot |f(x) - f^\star|^\theta \tag{12}$$

holds for every $x \in \mathbb{R}^n$ in a small enough open neighborhood of the stationary point $x = x^\star$. This inequality is guaranteed for analytic functions Łojasiewicz (1965). More precisely, when $x^\star$ is a local minimizer of f, the aforementioned relationship is explicitly given by

$$C = \left(\frac{p}{p-1}\right)^{\frac{p-1}{p}} \mu^{\frac{1}{p}}, \qquad \theta = \frac{p-1}{p}. \tag{13}$$

Therefore, analytic functions are always gradient dominated. However, while analytic functions are always smooth, smoothness is not required to attain gradient dominance.

The following results in Romero & Benosman (2020) summarized the finite-time convergence of q-RGF (equation 2) and q-SGF (equation 3) in continuous-time sense, which also motivates the main topic in this paper.

Theorem 1 (Romero & Benosman (2020))**.** *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and μ-gradient dominated of order $p \in (1, \infty)$ near a strict local minimizer $x^\star \in \mathbb{R}^n$. Let $c > 0$ and $q \in (p, \infty]$. Then, any maximal solution $x(\cdot)$, in the sense of Filippov, to the q-RGF (2) or q-SGF (3) will converge in finite time to $x^\star$, provided that $\|x(0) - x^\star\|_2 > 0$ is sufficiently small. More precisely, $\lim_{t\to t^\star} x(t) = x^\star$, where the convergence time $t^\star < \infty$ may depend on which flow is used, but in both cases is upper bounded by*

$$t^\star \leq \frac{\|\nabla f(x_0)\|_2^{\frac{1}{\theta}-\frac{1}{\theta'}}}{cC^{\frac{1}{\theta}}\left(1 - \frac{\theta}{\theta'}\right)}, \tag{14}$$

where $x_0 = x(0)$, $C = \left(\frac{p}{p-1}\right)^{\frac{p-1}{p}} \mu^{\frac{1}{p}}$, $\theta = \frac{p-1}{p}$, and $\theta' = \frac{q-1}{q}$. In particular, given any compact and positively invariant subset $S \subset \mathcal{D}$, both flows converge in finite time with the aforementioned convergence time upper bound (which can be tightened by replacing $\overline{\mathcal{D}}$ with S) for any $x_0 \in S$. Furthermore, if $\mathcal{D} = \mathbb{R}^n$, then we have global finite-time convergence, i.e. *finite-time convergence to any maximal solution (in the sense of Filippov) $x(\cdot)$ with arbitrary $x_0 \in \mathbb{R}^n$.*

In essence, the analysis introduced in Romero & Benosman (2020) consists of leveraging the gradient dominance to show that the energy function $\mathcal{E}(t) \triangleq f(x(t)) - f^\star$ satisfies the Lyapunov-like differential inequality $\dot{\mathcal{E}}(t) = \mathcal{O}(\mathcal{E}(t)^\alpha)$ for some $\alpha < 1$. The detailed proof is recalled in Appendix C for completeness.

[2]For more modern treatments in English, see Łojasiewicz & Zurro (1999); Bolte et al. (2007)

3 MAIN RESULTS: CONVERGENCE ANALYSIS FOR EULER DISCRETIZATION

With the previous discussion on continuous-time gradient flow, now we turn to the algorithm perspective. We propose the following general framework based on forward Euler discretization of the flows with finite-time convergence guarantee.

$$x_{k+1} = x_k + \eta F(x_k),\ \eta > 0 \tag{15}$$

where $F \in \{F_{q-\mathrm{RGF}}, F_{q-\mathrm{SGF}}\}$. We will show later, in Theorem 2, that the simple method leads, for small enough $\eta > 0$, to solutions that are $\mathcal{O}(\epsilon)$-close to the continuous-time flows. Also with a little abuse of notations, we will still use RGF/SGF to call the discrete-time algorithms induced by their corresponding continuous-time flows if the context is clear.

3.1 CLOSENESS OF THE DISCRETIZATION

We present here some convergence results of the proposed discretization. We first start with a proximity analysis, which aims at showing the closeness between the solutions of the continuous flows and the trajectories of their forward Euler discretization.

Theorem 2 (Closeness of Discretization)**.** *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable, locally L_f-Lipschitz, and μ-gradient dominated of order $p \in (1, \infty)$ in a compact neighborhood S of a strict local minimizer $x^\star \in \mathbb{R}^n$. Let $c > 0$ and $q \in (p, \infty]$. Then, for a given initial condition $x_0 \in S$, any maximal solution $x(t),\ x(0) = x_0$, (in the sense of Filippov) to the q-RGF given by (2) or the q-SGF flow (3), there exists an arbitrarily small $\epsilon > 0$ such that the solution x_k of any of the discrete algorithm (15), with sufficiently small $\eta > 0$, are ϵ-close to $x(t)$, i.e., $\|x_k - x(t)\|_2 \le \epsilon$ for $|t - k\eta| < \epsilon$, and s.t. the following bound holds*

$$f(x_k) - f(x^\star) \le L_f \epsilon + \left[(f(x_0) - f(x^\star))^{(1-\alpha)} - \tilde{c}(1-\alpha)\eta k \right]^{(1-\alpha)},\ L_f > 0,\ k \le k^\star, \tag{16}$$

where $\alpha = \frac{\theta}{\theta'},\ \theta = \frac{p-1}{p},\ \theta' = \frac{q-1}{q},\ \tilde{c} = c\left(\left(\frac{p}{p-1}\right)^{\frac{p-1}{p}} \mu^{\frac{1}{p}}\right)^{\frac{1}{\theta'}}$ *and* $k^\star = \frac{(f(x_0)-f(x^\star))^{(1-\alpha)}}{\tilde{c}(1-\alpha)\eta}$.

The analysis summarized in Theorem 2 is based on tools form hybrid control theory, and is detailed in Appendix D[3]. Theorem 2 shows that ϵ-convergence of $x_k \to x^\star$ can be achieved in a finite number of steps upper bounded by $k^\star = \frac{(f(x_0)-f(x^\star))^{(1-\alpha)}}{\tilde{c}(1-\alpha)\eta}$. This is a preliminary convergence result aiming to show the existence of discrete solutions obtained via the proposed discretization algorithms, which are ϵ-close to the continuous solutions of the finite-time flows. We also underline here that after x_k reaches an ϵ-neighborhood of $x^\star$, then $x_{k+1} \approx x_k,\ \forall k > k^\star$, since $x^\star$ is an equilibrium point of the continuous flows, i.e., $\Delta f(x^\star) = 0$, e.g. see Definition 2 in Appendix B.

3.2 CONVERGENCE RATES OF EULER DISCRETIZATION: GENERAL CASE

However, the bound (16) does not allow us to have a constructive practical value of the stepsize η, since the pair (ϵ, η) are implicitly related, for 'sufficiently small' η (refer to the definition of ϵ-closeness and the proof of Theorem 2 in Appendix D). To derive convergence rates of the proposed discretization with practical stepsize η, we need to introduce an extra assumption of L-Lipschitz smooth, as presented below.

Assumption 2 (Lipschitz Smoothness of Order q)**.** We assume the function f is L-Lipschitz smooth of order $q \in (1, 2]$, i.e., for any $x, y \in \mathbb{R}^n$,

$$\|\nabla f(y) - \nabla f(x)\|_2 \le L\|y - x\|_2^{q-1}. \tag{17}$$

Remark 2. The smoothness assumption is a very common setting in optimization algorithm literature (Nesterov, 2003; Beck, 2017). This assumption is also called as (L, q)-Hölder continuity (Devolder et al., 2014; Nesterov, 2015), it is easy to find that the condition will lead to the following property:

$$f(y) \le f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{q}\|y - x\|_2^q, \tag{18}$$

[3]Note that there might be several ways of approaching this proof. For instance, one could follow the general results on stochastic approximation of set-valued dynamical systems, using the notion of perturbed solutions to differential inclusions presented in Benaïm et al. (2005).

which is also called weak smoothness. When $q = 2$, the function will be Lipschitz smooth, so the above setting is a generalization of the common Lipschitz smoothness.

With these setting, now we will start our discussion from the general case, i.e., we have the access to the true gradient $\nabla f(x)$. The main result is presented in the following theorem, while we defer the proof to Appendix E. .

Theorem 3 (Convergence Rate of the general Case)**.** *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable, μ-gradient dominated of order q and L-Lipschitz smooth of order q following Assumption 1 and 2, then running q-RGF (equation 2 and equation 15) for K iterations with a stepsize $\eta = L^{\frac{1}{1-q}}$ satisfies that*

$$f(x_K) - f^\star \leq \left(1 - \kappa^{\frac{1}{1-q}}\right)^K (f(x_0) - f^\star), \tag{19}$$

and running q-SGF (equation 3 and equation 15) for K iterations with $\eta = (nL)^{\frac{1}{1-q}}$ satisfies that

$$f(x_K) - f^\star \leq \left(1 - \left(n^{\frac{q}{2}}\kappa\right)^{\frac{1}{1-q}}\right)^K (f(x_0) - f^\star), \tag{20}$$

where n is the dimension number and $\kappa \triangleq \frac{L}{\mu}$. The corresponding iteration complexity are $\mathcal{O}\left(\kappa^{\frac{1}{q-1}} \ln \frac{1}{\epsilon}\right)$ and $\mathcal{O}\left(\left(n^{\frac{q}{2}}\kappa\right)^{\frac{1}{q-1}} \ln \frac{1}{\epsilon}\right)$, respectively.

Remark 3. The above results show that with a constant stepsize, both the q-RGF and q-SGF can attain a linear convergence rate to get to the optimal point. Note that for the classical Lipschitz smooth case (i.e. $q = 2$), the RHS of the RGF result (equation 19) will reduce to $\left(1 - \kappa^{-1}\right)^K (f(x_0) - f^\star)$, which is the same as that in (Karimi et al., 2016, Theorem 1); while for SGF (equation 20), it will be $\left(1 - (n\kappa)^{-1}\right)^K (f(x_0) - f^\star)$, which recovers the result in (Beznosikov et al., 2020, Theorem 13)[4]. So we can conclude that our results extend the classical results in Lipschitz smoothness case.

3.3 Convergence Rates of Euler Discretization: Stochastic Case

Attaining the full gradient may be expensive in practical applications due to the large data size, while stochastic optimization algorithms are more applicable in such cases (Robbins & Monro, 1951; Bottou et al., 2018). So now we turn to discuss the complexity results when we have only the access to the unbiased estimator of the gradient. To formalize the discussion, we impose the following assumption on the gradient estimator, which is very common in stochastic optimization literature (Ghadimi & Lan, 2013; Bottou et al., 2018). .

Assumption 3 (Unbiased Gradient Estimator)**.** For any given $x \in \mathbb{R}^n$, we assume to have only the access of the unbiased gradient estimator, denoted as $g(x;\xi)$ where ξ is a random variable, while we assume that it satisfies that $\mathbb{E}_\xi[g(x;\xi)] = \nabla f(x),\ \mathbb{E}_\xi\|g(x;\xi) - \nabla f(x)\|_2^2 \leq \sigma^2$.

Stochastic q-RGF We present the convergence result of stochastic q-RGF in the following theorem, and defer the proof to Appendix F. For convenience, we denote $\psi \triangleq \frac{q}{q-1} \in [2, +\infty]$.

Theorem 4 (Convergence Rate of the Stochastic RGF)**.** *With the setting in Theorem 3, if we further assume for each $x \in \mathbb{R}^n$, we only have access to the unbiased estimator of the gradient in Assumption 3, then if we replace the gradient in q-RGF (equation 2) by its unbiased estimator $g(x) \triangleq \frac{1}{b(x)}\sum_{i=1}^{b(x)} g(x;\xi_i)$ with batch size $b(x) \in \mathbb{N}^+$. Then for q-RGF (equation 2 and equation 15) with $g(x)$, if we set $\eta = (\psi L)^{\frac{1}{1-q}}$, $b(x) \equiv \left(\frac{2\cdot(2\sigma)^\psi \mu^{\frac{1}{1-q}}}{\epsilon}\right)^{\frac{2}{\psi}}$, then we have*

$$\mathbb{E}[f(x_K) - f^\star] \leq \left(1 - \frac{2}{(2\psi)^\psi}\kappa^{\frac{1}{1-q}}\right)^K (f(x_0) - f^\star) + \frac{\epsilon}{2}. \tag{21}$$

Stochastic q-SGF The analysis of the Euler discretization of q-SGF in the stochastic case is a bit more complicated. Indeed, note that previously our discussion are based on ℓ_2-norm. Here, however, following the argument in Balles et al. (2020), we change the gradient dominance and Lipschitz smoothness assumption to hold in ℓ_∞-norm (and its dual norm ℓ_1).

[4]Note that $\frac{1}{n}\|\nabla f(x)\|_1^{-1}\operatorname{sign}(\nabla f(x))$ is an $\frac{1}{n}$-approximate compressor of gradient $\nabla f(x)$ (Karimireddy et al., 2019), so we can apply (Beznosikov et al., 2020, Theorem 13) to recover the result.

Assumption 4 (Gradient Dominance and Lipschitz Smoothness under (ℓ_1, ℓ_∞)-norms)**.** We assume the function f is μ-gradient dominated of order q and L-Lipschitz smooth of order q under ℓ_∞-norm (and its dual norm ℓ_1), i.e., for any $x, y \in \mathbb{R}^n$

$$\frac{q-1}{q}\|\nabla f(x)\|_1^{\frac{q}{q-1}} \geq \mu^{\frac{1}{q-1}}(f(x) - f^\star), \quad \|\nabla f(y) - \nabla f(x)\|_1 \leq L\|y - x\|_\infty^{q-1}. \tag{22}$$

Similarly the new smoothness condition above implies that $f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{q}\|y - x\|_\infty^q$ $(\forall\, x, y \in \mathbb{R}^n)$. And for the gradient estimator, we need the following assumption.

Assumption 5 (SPB Unbiased Gradient Estimator (Safaryan & Richtárik, 2021))**.** For any given $x \in \mathbb{R}^n$, we assume to have only the access of the unbiased gradient estimator under ℓ_1-norm, i.e., $\mathbb{E}_\xi[g(x;\xi)] = \nabla f(x)$, $\mathbb{E}_\xi\|g(x;\xi) - \nabla f(x)\|_1^2 \leq \sigma^2$; and for its mini-batch version $g(x)$ with batch size $b(x)$, we further assume that it satisfies *success probability bounds (SPB) property*, i.e., for each component of $g(x)$ and $\nabla f(x)$ (denoted as $g_i(x)$ and $\nabla_i f(x)$), we have $\exists\, p^* \in (\frac{1}{2}, 1]$ such that

$$\mathbb{P}(\text{sign}\,(\nabla_i f(x)) = \text{sign}\,(g_i(x))) \geq p^* > \frac{1}{2}, \quad \forall\, i \in \{1, 2, \cdots, n\}. \tag{23}$$

Remark 4. Note that similar assumptions under ℓ_∞-norm (and its dual norm ℓ_1) are widely adapted in the literature of sign-based algorithms, e.g., Karimi et al. (2016); Bernstein et al. (2018a;b); Balles et al. (2020). Here the extra SPB assumption, proposed in Safaryan & Richtárik (2021), is very important in the convergence analysis of stochastic sign-based algorithms. An intuitive understanding to justify SPB is that, with a large batch size, the output estimator $g(x)$ will be very similar to $\nabla f(x)$, so the corresponding probability above should be higher. For more details on the rationality of the assumption, we refer the reader to Safaryan & Richtárik (2021) for further discussions.

With the above preparation, we summarize the result of stochastic SGF in the following theorem, while deferring the proof to Appendix G.

Theorem 5 (Convergence Rate of the Stochastic SGF)**.** *Suppose the function f satisfies the setting in Assumption 4 above, then for q-SGF (equation 3 and equation 15) with an unbiased gradient estimator $g(x)$ of batch size $b(x) \in \mathbb{N}^+$, if we assume the gradient estimator $g(x)$ satisfies Assumption 5, and set $\eta = \left(\frac{L2^{\psi-1}}{2p^*-1}\right)^{\frac{1}{1-q}}$ and $b(x_k) \equiv \left(\frac{2\mu^{\frac{1}{1-q}}}{q\epsilon}\right)^{\frac{2}{\psi}}(2p^* - 1)^{-4}\sigma^2$, it satisfies that*

$$\mathbb{E}[f(x_K) - f^\star] \leq \left(1 - (2p^* - 1)^\psi\left(\kappa 2^{\psi-1}\right)^{\frac{1}{1-q}}\right)^K (f(x_0) - f^\star) + \frac{\epsilon}{2}. \tag{24}$$

Remark 5. The results in Theorem 4 and Theorem 5 shows that both stochastic RGF and SGF will drive the function value to linearly converge to the $\mathcal{O}(\epsilon)$-neighborhood of the optimal value. As a comparison on stochastic RGF, Lei et al. (2019) studied the convergence guarantee of stochastic gradient descent (SGD), which can be viewed as stochastic 2-RGF, under a similar setting. Also for stochastic SGF, we notice that recently Li et al. (2021) provided a similar convergence result of stochastic SGF in the classical setting (i.e., $q = 2$), their result also requires an lower bound on the probability of estimation correctness, which corresponds to the SPB assumption above. Our result can be viewed as an extension of their work.

4 Numerical Experiments

In this section, we will use several experiments to verify the effectiveness of our proposed algorithms, more details will be presented in Appendix H.

4.1 Numerical Experiments on Academic Examples

Let us show first on a simple numerical example that the acceleration in convergence, proven in continuous time for certain range of the hyperparmeters, can translate to some convergence acceleration in discrete time.

Example 1 Consider the following function: $f : \mathbb{R} \to \mathbb{R}$ and $f(x) \triangleq \frac{1}{q+1}|x|^{q+1}$, which can be shown to be q^q-gradient dominated of order $(q + 1)$. The test result of the proposed RGF compared

with classical GD is presented in Figure 1 below (with $q = 5$). The figure shows both sensitivity and performances of algorithms, we can find that q-RGF and q-SGF, comparing to the classical one ($q = 2$) and the over-large one ($q = 10$), can accommodate larger stepsize and attain better performances with the best-tuned stepsize.

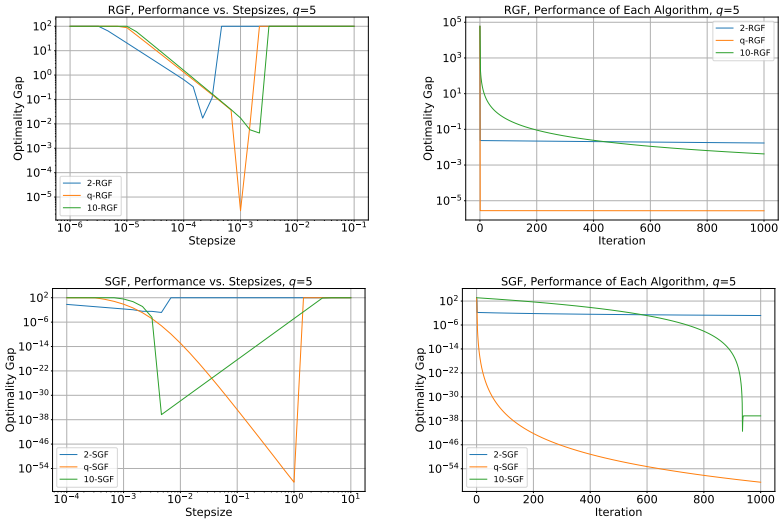

Figure 1: Results of Discretization of RGF and SGF with Different q on Example 1

Example 2 We consider the Rosenbrock function $f : \mathbb{R}^2 \to \mathbb{R}$, given by $f(x_1, x_2) = (a - x_1)^2 + b(x_2 - x_1^2)^2$, with parameters $a, b \in \mathbb{R}$, which is locally Lipschitz smooth. This function admits exactly one stationary point $(x_1^\star, x_2^\star) = (a, a^2)$ for $b \geq 0$, and is locally strongly convex, hence locally satisfies gradient dominance of order $p = 2$, which allows us to select $q > 2$ in q-RGF and q-SGF to achieve finite-time convergence in conitnuous-time. We report in Figure 2 the mean performance of forward-Euler discretization for q-RGF and q-SGF, with fixed step size[5], for several values of q, for 10 random initial conditions in $[0, 2]$. We observe that, as expected from the continuous flow analysis, for q close to 2, q-RGF behaves similar to GD in terms of convergence rate, whereas for $q > 2$ the finite-time convergence in continuous time translates to some acceleration in the associated discretization algorithm as well. Similarly for q-SGF, q closer to 2 translates to less accelerated algorithms, with a behavior similar to GD, whereas larger q values lead to accelerated convergence.

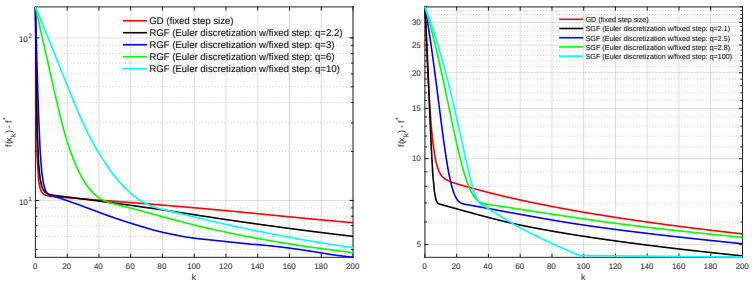

Figure 2: Results of Discretization of RGF and SGF with Different q on Example 2

4.2 Numerical Experiments on Real-world Data

We report here the results of our experiments using deep neural network (DNN) training on the SVHN dataset. Note that, we use PyTorch platform to conduct all the tests reported in this paper. Both CPU and GPU tests have been performed. The results reported in this section are CPU tests, whereas GPU tests are reported in Appendix H. We underline here that the DNNs are non-convex globally,

[5]We did multiple iterations to find the best step size for each algorithm (best values where between 10^{-4} and 10^{-2} depending on the algorithm). Details of the step size for each test are given in Appendix H.

however, one could assume at least local convexity, hence local gradient dominance of order $p = 2$, thus, we will select $q > 2$ in our experiments (see (Remark 6, Appendix H) for more explanations on the choice of q).

4.2.1 EXPERIMENTS ON SVHN DATASET

We test the proposed algorithms to train the same VGG16 CNN model with cross entropy loss on the SVHN dataset. We divided the dataset into a training set of 74 batches with 1000 images each, and a test set of 27 batches of 1000 images each, and ran 20 epochs of training over all the training batches. We tested the discretization of q-RGF ($c = 1,\ q = 2.1,\ \eta = 0.04$), and of q-SGF ($c = 10^{-3},\ q = 2.1,\ \eta = 0.04$) against Nesterov accelerated gradient descent (GD), and Adam[6]. Note from Figures 3, 4 it is clear that q-RGF and q-SGF give a good performance in terms of convergence speed, and final test performance 93%. We can also observe in Figure 4 that q-SGF, and q-RGF converge faster ($40\ min$ lead in average) than SGD and Adam for these tests, and reach an overall similar performance on the test-set. Additional numerical results, including GPU runs, can be found in Appendix H.

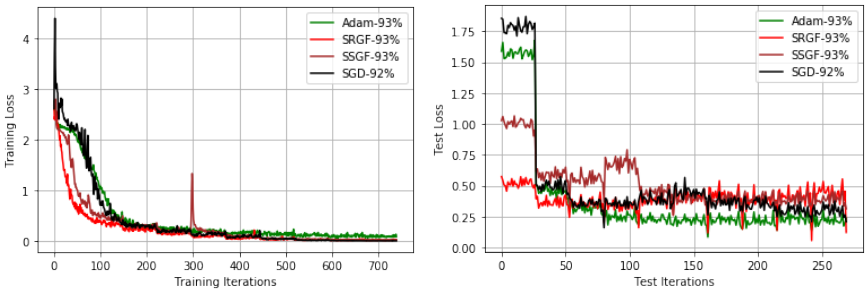

Figure 3: Losses for several optimization algorithms- SVHN: Train loss (left), test loss (right)

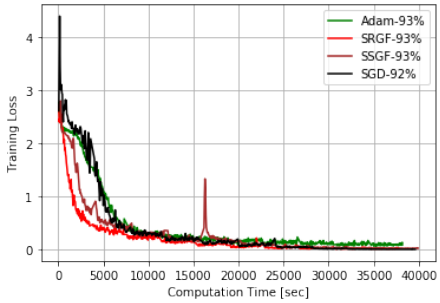

Figure 4: Training loss vs. computation time for several optimization algorithms- VGG-16- SVHN

5 CONCLUSION

We presetned some connections between optimization algorithms and continuous-time representations (dynamical systems) via discretization. We then reviewed two families of non-Lipschitz or discontinuous first-order optimization flows for continuous-time optimization, namely the q-RGF and q-SGF, whose distinguishing characteristic is their finite-time convergence. We then proposed a forward Euler discretization of these flows. Based on tools from hybrid systems control theory, we proved a closeness convergence bound for these algorithms, and then proposed several convergence rates in the deterministic and the stochastic setting. Finally, we conducted numerical experiments on a known deep neural network benchmark, which showed that the proposed discrete algorithms can outperform some state of the art algorithms, when tested on large DNN models.

[6]We also tested Adaptive gradient (AdaGrad), per-dimension learning rate method for gradient descent (AdaDelta), and Root Mean Square Propagation (RMSprop). However, since their performance was not competitive we decided not to report the plots to avoid overloading the figures.

REFERENCES

Hedy Attouch and Jerome Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming B*, 116(1):5–16, 2009.

Andrea Bacciotti and Francesca Ceragioli. Stability and stabilization of discontinuous systems and nonsmooth lyapunov functions. *ESAIM: Control, Optimisation and Calculus of Variations*, 4: 361–376, 1999.

Lukas Balles, Fabian Pedregosa, and Nicolas Le Roux. The geometry of sign gradient descent. *arXiv preprint arXiv:2002.08056*, 2020.

Amir Beck. *First-order methods in optimization*. SIAM, 2017.

Michel Benaïm, Josef Hofbauer, and Sylvain Sorin. Stochastic approximations and differential inclusions. *SIAM Journal on Control and Optimization*, 44(1):328–348, 2005.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pp. 560–569. PMLR, 2018a.

Jeremy Bernstein, Jiawei Zhao, Kamyar Azizzadenesheli, and Anima Anandkumar. signsgd with majority vote is communication efficient and fault tolerant. *arXiv preprint arXiv:1810.05291*, 2018b.

Aleksandr Beznosikov, Samuel Horváth, Peter Richtárik, and Mher Safaryan. On biased compression for distributed learning. *arXiv preprint arXiv:2002.12410*, 2020.

Jérôme Bolte, Aris Daniilidis, and Adrian Lewis. The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *Society for Industrial and Applied Mathematics*, 17:1205–1223, January 2007.

C.A Botsaris. A class of methods for unconstrained minimization based on stable numerical integration techniques. *Journal of Mathematical Analysis and Applications*, 63(3):729–749, 1978a.

C.A. Botsaris. Differential gradient methods. *Journal of Mathematical Analysis and Applications*, 63 (1):177–198, 1978b.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

R.W. Brockett. Dynamical systems that sort lists, diagonalize matrices and solve linear programming problems. In *IEEE Conference on Decision and Control*, pp. 799–803, 1988.

A.A. Brown. Some effective methods for unconstrained optimization based on the solution of systems of ordinary differential equations. *Journal of Optimization Theory and Applications*, 62 (2):211–224, August 1989.

Frank H. Clarke. Generalized gradients of lipschitz functionals. *Advances in Mathematics*, 40(1): 52–67, 1981.

Jorge Cortés. Finite-time convergent gradient flows with applications to network consensus. *Automatica*, 42(11):1993–2000, November 2006.

Jorge Cortés. Discontinuous dynamical systems. *IEEE Control Systems Magazine*, 28(3):36–73, June 2008.

Jorge Cortés and Francesco Bullo. Coordination and geometric optimization via distributed dynamical systems. *SIAM Journal on Control and Optimization*, 44(5):1543–1574, October 2005.

Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1):37–75, 2014.

M. Fazlyab, A. Koppel, V. M. Preciado, and A. Ribeiro. A variational approach to dual methods for constrained convex optimization. In *2017 American Control Conference (ACC)*, pp. 5269–5275, May 2017a.

M. Fazlyab, A. Koppel, V. M. Preciado, and A. Ribeiro. A variational approach to dual methods for constrained convex optimization. In *2017 American Control Conference (ACC)*, pp. 52690–5275, May 2017b.

M. Fazlyab, M. Morari, and V. M. Preciado. Design of first-order optimization algorithms via sum-of-squares programming. In *IEEE Conference on Decision and Control (CDC)*, pp. 4445–4452, December 2018.

Mahyar Fazlyab, Alejandro Ribeiro, Manfred Morari, and Victor M. Preciado. Analysis of optimization algorithms via integral quadratic constraints: Nonstrongly convex problems. *SIAM J. Optim*, 28(3):2654–2689, 2018.

Aleksei Fedorovich Filippov and F. M. Arscott. *Differential equations with discontinuous righthand sides*. Kluwer Academic Publishers Group, Dordrecht, Netherlands, 1988.

Guilherme Franca, Daniel Robinson, and Rene Vidal. Admm and accelerated admm as continuous dynamical systems. In *International Conference on Machine Learning*, pp. 1559–1567. PMLR, 2018.

G. França, D.P. Robinson, and R. Vidal. A dynamical systems perspective on nonsmooth constrained optimization. *arXiv preprint 1808.04048*, 2019a.

G. França, J. Sulam, D. Robinson, and R. Vidal. Conformal symplectic and relativistic optimization. *arXiv preprint 1903.04100*, 2019b.

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Uwe Helmke and John Barratt Moore. *Optimization and Dynamical Systems*. Springer-Verlag, 1994.

Qing Hui, Wassim Haddad, and Sanjay Bhat. Semistability, finite-time stability, differential inclusions, and discontinuous dynamical systems having a continuum of equilibria. *IEEE Transactions on Automatic Control*, 54:2465–2470, November 2009.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 795–811. Springer, 2016.

Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian Stich, and Martin Jaggi. Error feedback fixes signsgd and other gradient compression schemes. In *International Conference on Machine Learning*, pp. 3252–3261. PMLR, 2019.

Yunwen Lei, Ting Hu, Guiying Li, and Ke Tang. Stochastic gradient descent for nonconvex learning without bounded gradient assumptions. *IEEE transactions on neural networks and learning systems*, 31(10):4394–4400, 2019.

L. Lessard, B. Recht, , and A. Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM J. Optim*, 26(1):57–95, 2016.

Xiuxian Li, Kuo-Yi Lin, Li Li, Yiguang Hong, and Jie Chen. On faster convergence of scaled sign gradient descent. *arXiv preprint arXiv:2109.01806*, 2021.

S. Łojasiewicz. A topological property of real analytic subsets (in French). *Les équations aux dérivées partielles*, pp. 87–89, 1963.

S. Łojasiewicz. *Ensembles semi-analytiques*. Centre de Physique Theorique de l'Ecole Polytechnique, 1965. URL `https://perso.univ-rennes1.fr/michel.coste/Lojasiewicz.pdf`.

StanisŁaw Łojasiewicz and Maria-Angeles Zurro. On the gradient inequality. *Bulletin of the Polish Academy of Sciences, Mathematics*, 47, January 1999.

Michael Muehlebach and Michael Jordan. A dynamical systems perspective on Nesterov acceleration. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4656–4662, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Yu Nesterov. Universal gradient methods for convex optimization problems. *Mathematical Programming*, 152(1):381–404, 2015.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

A. Orvieto and A. Lucchi. Shadowing properties of optimization algorithms. In *Neural Information Processing Systems*, December 2019.

Bradley Paden and Shankar Sastry. A calculus for computing filippov's differential inclusion with application to the variable structure control of robot manipulators. *IEEE Transactions on Circuits and Systems*, 34:73–82, February 1987.

Boris Polyak. Gradient methods for the minimisation of functionals (in Russian). *USSR Computational Mathematics and Mathematical Physics*, 3:864–878, December 1963.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

O. Romero and M. Benosman. Finite-time convergence in continuous-time optimization. In *International Conference on Machine Learning*, Vienna, Austria, July 2020.

O. Romero, S. Chaterjee, and S. Pequito. Convergence of the expectation-maximization algorithm through discrete-time lyapunov stability theory. *Proceedings of the American Control Conference (ACC)*, pp. 163–168, July 2019.

Mher Safaryan and Peter Richtárik. Stochastic sign descent methods: New algorithms and better theory. In *International Conference on Machine Learning*, pp. 9224–9234. PMLR, 2021.

R. G. Sanfelice and A. R. Teel. Dynamical properties of hybrid systems simulators. *Automatica*, 46: 239–248, 2010.

Johannes Schropp. Using dynamical systems methods to solve minimization problems. *Applied Numerical Mathematics*, 18(1):321–335, 1995.

Johannes Schropp and I Singer. A dynamical systems approach to constrained minimization. *Numerical Functional Analysis and Optimization*, 21:537–551, May 2000.

D. Scieur, V. Roulet, F. Bach, , and A. d'Aspremont. Integration methods and optimization algorithms. In *Neural Information Processing Systems*, December 2017.

Bin Shi, Simon Du, Michael Jordan, and Weijie Su. Understanding the acceleration phenomenon via high-resolution differential equations. *arXiv preprint 1810.08907*, October 2018.

J.A. Snyman. A new and dynamic method for unconstrained minimization. *Applied Mathematical Modelling*, 6(6):448–462, December 1982.

J.A. Snyman. An improved version of the original leap-frog dynamic method for unconstrained minimization: LFOP1(b). *Applied Mathematical Modelling*, 7(3):216–218, June 1983.

Andrew M. Stuart and A. R. Humphries. *Dynamical systems and numerical analysis*. Cambridge University Press, first edition, November 1998.

W. Su, S. Boyd, and E. J. Candes. A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems*, pp. 2510–2518. Curran Associates, Inc., 2014.

Adrien Taylor, Bryan Van Scoy, and Laurent Lessard. Lyapunov functions for first-order methods: Tight automated convergence guarantees. In *International Conference on Machine Learning*, Stockholm, Sweden, July 2018.

J. Wang and N. Elia. A control perspective for centralized and distributed convex optimization. In *IEEE Conference on Decision and Control and European Control Conference*, pp. 3800–3805, December 2011.

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.

A. Wilson. *Lyapunov Arguments in Optimization*. PhD thesis, UC Berkeley, 2018.

Ashia Wilson, Lester Mackey, and Andre Wibisono. Accelerating rescaled gradient descent: Fast optimization of smooth functions. *arXiv preprint arXiv:1902.08825*, 2019.

Abbas K. Zghier. *The use of differential equations in optimization*. PhD thesis, Loughborough University, 1981.

J. Zhang, A. Mokhtari, S. Sra, , and A. Jadbabaie. Direct runge-kutta discretization achieves acceleration. In *Neural Information Processing Systems*, December 2018.