ODE Discretization Schemes as Optimization Algorithms

Romero, Orlando; Benosman, Mouhacine; Pappas, Geroge

TR2022-159 December 09, 2022

Abstract

Motivated by the recent trend in works that study optimization algorithms from the point of view of dynamical systems and control, we seek to understand how to best systematically discretize a given generic continuous-time analogue of a gradient-based optimization algorithm, represented by ordinary differential equations (ODEs). To this end, we show how a suboptimality bound for such continuous-time algorithms can be combined with an ODE solver's accuracy bound in order to provide non-asymptotic suboptimality bounds upon discretization. In particular, we show that subexponential, exponential, and finite-time convergence rates in continuous time can be nearly matched upon discretization by merely using off-the-shelf ODE solvers of modestly high order. We then illustrate our findings on a modified version of the celebrated second-order ODE that models Nesterov's accelerated gradient. Lastly, we apply our approach on the rescaled gradient flow.

IEEE Conference on Decision and Control (CDC) 2022

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

ODE Discretization Schemes as Optimization Algorithms

Orlando Romero[1,⋆], Mouhacine Benosman[2], and George J. Pappas[1]

Abstract— **Motivated by the recent trend in works that study optimization algorithms from the point of view of dynamical systems and control, we seek to understand how to best systematically discretize a given generic continuous-time analogue of a gradient-based optimization algorithm, represented by ordinary differential equations (ODEs). To this end, we show how a suboptimality bound for such continuous-time algorithms can be combined with an ODE solver's accuracy bound in order to provide non-asymptotic suboptimality bounds upon discretization. In particular, we show that subexponential, exponential, and finite-time convergence rates in continuous time can be nearly matched upon discretization by merely using off-the-shelf ODE solvers of modestly high order. We then illustrate our findings on a modified version of the celebrated second-order ODE that models Nesterov's accelerated gradient. Lastly, we apply our approach on the rescaled gradient flow.**

I. Introduction

Nonlinear optimization plays a major role in many scientific, engineering, and otherwise applied areas, such as economics, machine learning, and control. While it is typically preferable to design optimization algorithms that are finely tuned to the problem at hand, there are areas, such as in deep learning, where cost functions may be too unwieldy or unstructured for overly specialized optimization algorithms to be applied. For this and other reasons, there is still significant value in continuing to purse the design and analysis of general-purpose optimization algorithms.

As it turns out, many gradient-based optimization algorithms can be seen directly or well-approximated by discretizations of continuous-time dynamical systems modeled by ODEs, e.g. [1], [2], [3], [4]. Designing and analyzing optimization algorithms in continuous time via ODEs is attractive to many researchers due to their often easier geometric interpretation, analytical handling, and overall amount of prior literature that exists on stability properties of ODEs, e.g., [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]. In particular, asymptotic Lyapunov stability of state-space dynamical systems, while most commonly associated with the control theory community (e.g., [16]), offers a vast array of tools to study how the trajectories of ODEs converge to certain points or regions in the state space.

Interestingly, the majority of the aforementioned references appear to use continuous-time representations of optimization algorithms almost exclusively to aid in the design and analysis, but ultimately the only thing that is explicitly carried over upon discretization is the intuition gained in continuous time, with everything else manually redone. For instance, Lyapunov functions that worked in continuous time can often be used as the basis for a similar Lyapunov function in discrete time, with only minor adjustments. However, some properties in continuous time are not guaranteed to be preserved upon discretization unless some care is taken.

In this work, we address the aforementioned gap by building upon the ideas in [1], [17]. Scieur et al [1] studied the potential of the *linear multistep method* of discretization, when applied to the *gradient flow* ODE. On the other hand, Zhang et al [17] studied explicit *Runge-Kutta discretizations* of a modified version of the second-order ODE that models Nesterov's accelerated gradient [8], first studied in [9]. We follow up on [1], [17] by considering *generic ODEs that represent gradient-based dynamical systems and study the optimization performance for off-the-shelf ODE solvers applied to such systems.*

Our main contributions are as follow. First, we propose a formalism that captures the following problems: *(i)* how to discretize a continuous-time optimization algorithm to satisfy an arbitrary suboptimality tolerance? *(ii)* given a convergence rate for the continuous-time algorithm, what kind of corresponding rates can we ensure upon discretization? In particular, what are the fundamental limits and can we ensure matching rates?

More concretely, we prove the following novel near-matching rates: 1. if ε-optimality can be achieved in continuous time at a subexponential rate $\mathcal{O}(\varepsilon^{-\frac{1}{\delta}})$ (length of interval of integration), then ε-optimality can be achieved upon discretization at a rate $\mathcal{O}(\varepsilon^{-\left(\frac{1}{\delta}+\frac{1}{\nu}\left(1+\frac{1}{\delta}\right)\right)})$ (number of time steps), when employing an ODE solver of accuracy order $\mathcal{O}(h^\nu)$; 2. If the continuous-time algorithm converges at an exponential rate $\mathcal{O}\left(\log\left(\frac{1}{\varepsilon}\right)\right)$, then we can ensure a corresponding rate $\mathcal{O}\left(\log\left(\frac{1}{\varepsilon}\right)\varepsilon^{-\frac{1}{\nu}}\right)$ upon discretization; 3. if the continuous-time algorithm converges with a finite-time rate $\mathcal{O}(1)$, then we can ensure a corresponding rate $\mathcal{O}(\varepsilon^{-\frac{1}{\nu}})$ upon discretization. In all 3 cases, taking $\nu \to \infty$ (i.e. using high-order schemes) leads heuristically to matching the continuous-time rates.

As particular case study, we first consider the modified version of Nesterov's ODE [8] proposed by [9], which depends on a parameter $p > 2$ related to time dilation. Since the continuous-time algorithm in question can be proved to converge at a rate $\mathcal{O}(1/\varepsilon^{\frac{1}{p}})$, we can then establish a novel discretized rate $\mathcal{O}(1/\varepsilon^{\frac{1}{p}+\frac{1}{\nu}\left(1+\frac{1}{p}\right)}) = \mathcal{O}(1/\varepsilon^{\frac{1}{p}\frac{p+\nu+1}{\nu}})$. When contrasted to the rate $\mathcal{O}(1/\varepsilon^{\frac{1}{p_{\max}}\frac{\nu+1}{\nu}})$ derived in [17], where $p_{\max}$ denotes the largest p that will allow for a certain flatness condition to hold (required in [17] but not in this work), we

[1] Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA.

[2] Mitsubishi Electric Research Laboratories, Cambridge, MA, USA.

[⋆] Corresponding author. E-mail: oromero@seas.upenn.edu.

can see that we are able to outperform it by tuning p and ν appropriately (e.g., $\max(p,\nu) \geq p_{\max}$ suffices).

As a second case study, we consider the *rescaled gradient flow*, which was also proposed first in [9]. In particular, we note that it has a continuous-time rate $\mathcal{O}(1/\varepsilon^{\frac{1}{q-1}})$, with $q > 1$ a parameter also related to time dilation. Therefore, we can guarantee a nearly matching rate $\mathcal{O}(1/\varepsilon^{\frac{1}{q-1}\frac{q+\nu}{\nu}}) \approx \mathcal{O}(1/\varepsilon^{\frac{1}{q-1}})$. As pointed out in [9], the rate $\mathcal{O}(1/\varepsilon^{\frac{1}{q-1}})$ can be matched *exactly* upon discretization for integer q. However, unlike our purely gradient-based approach, the exact matching rate described in [9] requires exact oracle access to the first q derivatives of the cost function.

For the aforementioned case studies (dropping the gradient dominance of order $p = q$ assumption), taking $p, q \to \infty$ leads to the heuristic rate $\mathcal{O}(1/\varepsilon^{\frac{1}{\nu}})$, which is remarkably close to the known upper bound $\mathcal{O}(1/\varepsilon^{\frac{1}{\nu-1}})$ and lower bound $\Omega(1/\varepsilon^{\frac{2}{3\nu+1}})$ attainable with oracle access to the first ν derivatives of the cost function

In the paper [18] that followed-up [9], [17], it was proved that the rate $\mathcal{O}(1/\varepsilon^{\frac{1}{q-1}})$ can be matched exactly for convex functions without requiring high-order derivatives of the cost function, provided that a strong smoothness condition holds. Further, that same paper showed that, for uniformly convex functions of order $p = q$ (which implies gradient dominance of order p), a linear convergence rate is attained. We recover this result without requiring strong smoothness nor uniform convexity, instead merely gradient dominance of order $p = q$. More precisely, under such conditions we show a nearly matching rate $\mathcal{O}\left(\log\left(\frac{1}{\varepsilon}\right)\frac{1}{\varepsilon^{\frac{1}{\nu}}}\right) \approx \mathcal{O}\left(\log\left(\frac{1}{\varepsilon}\right)\right)$.

Structure of the paper

In Section II, we formalize the problem statement. In Section III, we first review some key concepts from the numerical ODE literature and then present our proposed solution to the problem statement. In Section IV, we illustrate our findings by applying them to a parameterized family of continuous-time optimization algorithms, namely the rescaled gradient flow. In Section V, we summarize our findings and future work directions.

II. Problem Formulation

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function that we seek to minimize and let us assume throughout that f is continuously differentiable and has non-empty solution set $\mathcal{X}^\star = \arg\min f$. Suppose that we have designed a dynamical system

$$\dot{X}(t) = F(t, X(t)) \tag{1a}$$

$$x(t) = G(X(t)) \tag{1b}$$

that serves as a *continuous-time* analogue of a gradient-based algorithm to minimize f. More precisely, $X(t)$ serves as the state of the system and $x(t)$ as the output, which ought to converge towards a minimizer.

Example 1 (Gradient flow)**.** The *gradient flow* $\dot{x} = -\nabla f(x)$ can be proved to converge towards a (possibly local) minimizer under mild conditions. For instance, if f is convex in an open convex neighborhood of a strict local minimizer $x^\star$, then indeed $x(t) \to x^\star$ will hold, provided that the initial condition $x(t_0) = x_0$ is sufficiently near $x^\star$. In particular, $f(x(t)) - f(x^\star) \leq \frac{\|x_0 - x^\star\|^2}{(f(x_0) - f(x^\star))^{-1} + t} = \mathcal{O}(1/t)$. To quantify the convergence rate of $x(t) \to x^\star$, more regularity is required, as we will see in Section IV.

Example 2 (Nesterov's ODE, [8])**.** The gradient-based system described by the second-order ODE

$$\ddot{x} + \frac{3}{t}\dot{x} + \nabla f(x) = 0 \tag{2}$$

with initial conditions $x(0) = x_0$ and $\dot{x}(0) = 0$ was proposed in [8] as a natural continuous-time interpretation of Nesterov's accelerated gradient (NAG) algorithm. Naturally, (2) can be rewritten in a state-space form (1), for instance, by taking $X = \begin{bmatrix} x & \dot{x} \end{bmatrix}^\top$. The original authors in [8] proved that (2) achieves a convergence rate $f(x(t)) - f(x^\star) = \mathcal{O}(1/t^2)$, but the convergence rate for $\|x(t) - x^\star\|$ was only recently established in [12] for a generalization of (2).

Before we proceed, let us be more precise about what is permissible as a gradient-based continuous-time optimization algorithm. In what follows, $\mathbf{O}_g$ will denote a generic oracle for a function g, i.e. a black box algorithm that evaluates g at any desired input in its domain.

Assumption 1 (Gradient-based systems)**.** The functions $F : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$ and $G : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^n$ may only depend on f via oracles $\mathbf{O}_f$ and $\mathbf{O}_{\nabla f}$ that may only be queried finitely many times per evaluation of F and G. Further, the output $x(t)$ of (1) converges to a (possibly local) minimizer of f as $t \to \infty$.

In order to implement a continuous-time algorithm in practice, some form of *discretization* is a necessity due to the digital nature of modern computers. To this end, let us assume that some finite-memory iterative ODE solver is applied to (1) on a time interval $[t_0, T]$ partitioned as $t_0 < \ldots < t_N = T$. More precisely, setting $h_k := t_{k+1} - t_k$ (known as the *step sizes*), we seek to construct approximations X_k of $X(t_k)$ via a recursive scheme that may require solving a system of (possibly nonlinear) auxiliary equations. Inspired by the *general linear method* of J.C. Butcher [19], we consider general *multistep-multistage* schemes that can be written as

$$X_{k+1} = \Phi_{h_k}(\mathbf{t}_k, \mathbf{X}_k, \mathbf{Y}_k) \tag{3a}$$

$$\mathbf{Y}_k \in \Psi_{h_k}(\mathbf{t}_k, \mathbf{X}_k, \cdot)^{-1}(0) \tag{3b}$$

where $\mathbf{t}_k = (t_k, \ldots, t_{k-r+1})$, $\mathbf{X}_k = (X_k, \ldots, X_{k-r+1})$, and $\mathbf{Y}_k = (Y_{k,1}, \ldots, Y_{k,s})$, with $X_k, Y_{k,i} \in \mathbb{R}^m$. The components $Y_{k,1}, \ldots, Y_{k,s}$ of the the auxiliary variable $\mathbf{Y}_k$ are called the *stages* of the scheme and depict intermediate steps given by solving the auxiliary equation $\Psi_{h_k}(\mathbf{t}_k, \mathbf{X}_k, \mathbf{Y}) = 0$ in $\mathbf{Y}$. The values $X_0, \ldots, X_{p-1}$ correspond to initial approximations for $X(t_0), \ldots, X(t_{p-1})$. Since $\mathbf{Y}_k = \mathbf{Y}_k(h_k, \mathbf{t}_k, \mathbf{X}_k)$, we may omit it from $\Phi_{h_k}(\mathbf{t}_k, \mathbf{X}_k, \mathbf{Y}_k)$, whenever the auxiliary equation does not require emphasis.

Definition 1. The scheme (3) is said to be *explicit* (w.r.t. to a given oracle $\mathbf{O}$) if the equation $\Psi_h(\mathbf{t}, \mathbf{X}, \mathbf{Y}) = 0$

can be solved in $\mathbf{Y}$ exactly and in finitely many arithmetic operations and queries to $\mathbf{O}$. Otherwise, the scheme is said to be *implicit* (w.r.t. $\mathbf{O}$).

In almost all practical situations, $\mathbf{O}$ will simply be an oracle for F, but in principle we could consider related observables or even *multiderivative* methods, which query derivatives of F (e.g. Taylor and Obreshkov methods). In the context of this work, relevant additional observables could be queries to the cost function to guide the construction of stepsizes in a similar fashion to backtracking inexact line search.

Example 3 (Euler's method)**.** Perhaps the simplest discretization scheme is the Euler method, based on a first-order Taylor approximation of $X(t)$. In particular, approximating $X(t_{k+1})$ around $t = t_k$ with ansatz $X_k \approx X(t_k)$, we end up with the scheme

$$X_{k+1} = X_k + h_k F(t_k, X_k), \tag{4}$$

whereas approximating $X(t_k)$ around $t = t_{k+1}$, we end up with the scheme (after reorganizing terms)

$$X_{k+1} = X_k + h_k F(t_{k+1}, X_{k+1}), \tag{5}$$

and these are, respectively, referred to as the *forward* (or *explicit*) and *backward* (or *implicit*) Euler methods.

Example 4 (Taylor series method)**.** The natural improvement and generalization over the Euler method is to take higher-order Taylor approximations. For instance, for the forward approximation of order p, we end up with the scheme

$$X_{k+1} = X_k + \sum_{j=1}^{p} \frac{h_k^j}{j!} F^{(j-1)}(t_k, X_k), \tag{6}$$

where $F^{(j-1)}$ denotes the $(j-1)$-th Lie derivative of F w.r.t. $\dot{X} = F(t, X)$, i.e. $F^{(j)}(t, X) = \frac{\partial F^{(j-1)}}{\partial t}(t, X) + \frac{\partial F^{(j-1)}}{\partial X}(t, X) F(t, X)$, with $F^{(0)}(t, X) = F(t, X)$, where $\frac{\partial}{\partial X}$ denotes the usual Jacobian matrix operator.

While high-order Taylor series methods offer attractive properties at the theory level, their main downside is that they require *exact* access to high-order derivatives of $F(t, X)$, which is often impractical. In the following two examples, we describe two widely popular methods that attempt to circumvent the issue highlighted for Taylor series methods by linear approximation. Both of the following schemes are particular cases of the *general linear method*, originally proposed by J.C. Butcher in 1966 [20].

Example 5 (Linear multistep method)**.** The linear multistep method (LMM) stores (finitely many) past iterates and queries of F and linearly combines them as an approximation of the net sum of high-order Taylor terms:

$$X_{k+r} + \sum_{j=0}^{r-1} a_j X_{k+j} = h_{k+r} \sum_{j=0}^{r} b_j F(t_{k+j}, X_{k+j}), \tag{7}$$

where a_i, b_i are tunable parameters. Note that the scheme is explicit if and only if $b_r = 0$. Further, it is a 1-stage method.

Example 6 (Runge-Kutta method)**.** The *Runge-Kutta* (RK) method opts to trade-off the memory overhead of LMM by instead performing more computations per time step and querying F multiple times. More precisely, F is queried multiple times per time step in order to find intermediate iterates that are then linearly combined into a single update:

$$X_{k+1} = X_k + h_k \sum_{i=1}^{s} b_i F(t_k + c_i h_k, Y_{k,i}) \tag{8a}$$

$$Y_{k,i} = X_k + h_k \sum_{j=1}^{s} a_{ij} F(t_k + c_j h_k, Y_{k,j}), \tag{8b}$$

where a_{ij}, b_i, c_j are tunable parameters. Since these parameters fully characterize a Runge-Kutta method, we may identify the scheme as simply (A, b, c), though it is customary to express them in the so-called *Butcher tableau* form.

Note that the scheme is explicit if and only if $A = (a_{ij})$ is strictly lower triangular. In that case, the scheme requires at most $\frac{1}{2}s(s+1) = \mathcal{O}(s^2)$ queries of F. For implicit schemes, however, infinitely many queries would be theoretically required, but in practice the auxiliary nonlinear equations would be solved in finitely many steps and queries, up to a given error tolerance. Regardless, a major advantage of performing more expensive steps and querying F multiple times per step is that Runge-Kutta schemes have the potential of taking significantly larger step sizes, and thus fewer step sizes overall, without compromising accuracy.

In this work, we are primarily interested in gradient-based optimization algorithms seen as discretizations of ODEs that serve as continuous-time analogues of gradient-based optimization algorithms. For this reason, we make the following assumption:

Assumption 2 (Gradient-based discretization)**.** The functions $\Phi_h(\cdot), \Psi_h(\cdot)$ (with arbitrary $h > 0$) and the construction of the step sizes $\{h_k\}$ may only depend on f and F via oracles $\mathbf{O}_f, \mathbf{O}_{\nabla f}, \mathbf{O}_F$. Further, the stepsizes $\{h_k\}$ must be constructed recursively.

Before we formalize our problem statement, let us make a regularity assumption that allows us to provide global convergence guarantees without requiring convexity:

Assumption 3 (Invexity)**.** There exists some vector-valued function $\xi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ such that $f(y) - f(x) \geq \langle \xi(x, y), \nabla f(x) \rangle$ for all $x, y \in \mathbb{R}^n$.

It is not difficult to see that *invex* functions (i.e. those satisfying Assumption 3) are exactly those whose stationary points are all global minimizers.

With this, we can now formulate our problem statement. But first, recall that $\alpha : [0, r) \to \mathbb{R}$ with $0 < r \leq \infty$ is said to be of *class* $\mathcal{K}$ if it is continuous, strictly increasing, and satisfies $\alpha(0) = 0$. Further, $\alpha : [0, r) \times [0, \infty) \to \mathbb{R}$ is said to be of *class* $\mathcal{KL}$ if $\alpha(\cdot, t)$ is of class $\mathcal{K}$ and $\alpha(s, \cdot)$ is continuous, non-negative, non-decreasing, and such that $\alpha(s, t) \to$ as $t \to \infty$.

Problem 1. Consider a class $\mathcal{F}$ of continuously differentiable differentiable cost functions $f : \mathbb{R}^n \to \mathbb{R}$ of interest satisfying Assumption 3, as well as a continuous-time gradient-based algorithm (1) satisfying Assumption 1 for every $f \in \mathcal{F}$. Consider some performance metric $V(x)$ that is positive semidefinite w.r.t. $x = x^\star$, such as $f(x) - f(x^\star)$, $\|\nabla f(x)\|^2$, or $\|x - x^\star\|^2$. Given some optimality error tolerance $\varepsilon > 0$, we seek to design a partition $\{t_0 < \ldots < t_N = T\}$ and a discretization scheme (3) satisfying Assumption 2, such that $x_k := G(X_k)$ satisfies $V(x_N) \leq \varepsilon$. In particular, given a non-asymptotic convergence rate in continuous time that holds for all $f \in \mathcal{F}$, of the form $V(x(T)) \leq \alpha(V(x(t_0)), T - t_0)$ with α a class $\mathcal{KL}$ function, we seek the tightest corresponding rate upon discretization $V(x_N) \leq \beta(V(x_0), N)$, with β a class $\mathcal{KL}$ function.

III. Combining Continuous-Time Convergence and ODE Solver Accuracy

Conceptually, ODE solvers are designed to approximate the overall solution curves $X(t)$ of an ODE $\dot{X} = F(t, X)$. Therefore, as we increasingly partition the time interval $[t_0, T]$, we expect $X_k \approx X(t_k)$, with convergence in the limit $h := \|(h_k)\|_\infty \to 0$. In this section, we first (Sec. III-A) make this notion more precise by borrowing some useful definitions from the numerical ODE literature, slightly adapted for our purposes. Then (Sec. III-B), we propose a method to combine the accuracy order of an ODE solver with the convergence rate of the given continuous-time gradient-based algorithm. This way, we can provide iteration complexity guarantees for the discretization now viewed as an optimization algorithm.

A. ODE Discretization Preliminaries: Truncation Error, Consistency, Convergence

The following two definitions capture the notion of $X_k \approx X(t_k)$ for a scheme (3) as the time interval $[t_0, T]$ is progressively more finely partitioned.

Definition 2 (Local error)**.** The *local truncation error* of scheme (3) is the error accrued in a single step when starting from a perfect approximation, i.e. $\tau_k := \Phi_{h_k}(\mathbf{t}_k, X(t_k), \ldots, X(t_{k-r+1})) - X(t_{k+1})$. We say that the scheme is *consistent* if the $\|\tau_k\|/h_k \to 0$ as $h_k \to 0$, for all smooth F. In particular, if $\|\tau_k\| \leq h_k \omega(h_k)$ holds for all small enough $h_k > 0$, for some function $\omega(h)$ of class $\mathcal{K}$, we call $\omega(h)$ a *(local) accuracy order function.*

Definition 3 (Global error)**.** The *global truncation error* of scheme (3) is the accumulated error accrued over the discretization process, i.e. $\mathcal{T}_k := X_k - X(t_k)$. We say that the scheme is *convergent* if there exists a starting procedure $\mathbf{X}_0 = \mathbf{X}_0(h_0, \ldots, h_{r+1})$ such that $\max_k \|\mathcal{T}_k\| \to 0$ as $h := \max_k h_k \to 0$, for all smooth F. In particular, if $\max_k \|\mathcal{T}_k\| \leq K(T - t_0)\omega(h)$ holds for all small enough $h > 0$, for some functions $K(\cdot), \omega(\cdot)$ of class $\mathcal{K}$, we call $\omega(h)$ a *(global) accuracy order function.*

Next, we exemplify the notions above and briefly summarize how they can be (partially) characterized for the forward Euler method and the Runge-Kutta method. For the linear multistep method, [1] summarizes well the convergence and stability properties.

1) Forward Euler's method: Consider the forward Euler method (4). Suppose that $F(t, X)$ is continuous in t, twice continuously differentiable in X, and has first and second-order partial derivatives in X bounded by $L > 0$ and $M > 0$, respectively. Then, if the stepsizes are bounded by $h_k \leq h$, it follows that $\|\tau_k\| \leq \frac{M}{2} h_k^2$ and $\|\mathcal{T}_k\| \leq \frac{M}{2L}(e^{L(T-t_0)} - 1)h$ (see [21, §2.1]).

2) Runge-Kutta's method: Consider the s-stage Runge-Kutta method (8). It is not difficult to see that the scheme is consistent if and only if $\sum_{i=1}^{s} b_i = 1$. Common approaches to construct high-order Runge-Kutta schemes are based, for instance, on the so-called *simplifying conditions* $B(\rho), C(\eta), D(\xi)$, as well quadrature-based methods and collocation methods [22, p. 237-238]. Unsurprisingly, the local accuracy order carries over to the global error. More precisely, if $F(t, \cdot)$ is L-Lipschitz and the Runge-Kutta scheme (A, b, c) satisfies $\|\tau_k\| \leq Ch^{\nu+1}$, then $\|\mathcal{T}_k\| \leq \frac{C}{L}(e^{L(T-t_0)} - 1)h^\nu$ [22, Thm. 3.4.7]. While the exponential growth $e^{L(T-t_0)}$ is initially worrisome, and was specifically addressed for the gradient flow in [23] by exploiting hyperbolicity, it turns out that tamer bounds for $K(\cdot)$ can be attained under Lyapunov stability. More precisely, for non-expansive and contractive systems, respectively, $K(\cdot)$ can be proved to be bounded and exponentially decaying [24]. For asymptotically and exponentially stable systems, respectively, $K(\cdot)$ can be proved to grow linearly at most and be bounded [25]. Explicit Runge-Kutta schemes of order $\mathcal{O}(h^\nu)$ can be constructed with $s = \nu$ stages, for $\nu \leq 4$, but for $\nu > 4$, a fundamental limit of $s > \nu$ is attained. While the smallest number of stages required to attain order $\mathcal{O}(h^\nu)$ with general $\nu > 4$ is unknown, we can construct schemes of such order using at most $s = \lceil \frac{3\nu^2}{8} \rceil$ stages [21, §3.2.4].

B. Main results

Let us now see how we can provide a partial answer to Problem 1. But first, recall that a class $\mathcal{K}$ function $\phi : [0, \infty] \to [0, \infty]$ is a *modulus of continuity* of a continuous function $g : \mathcal{X} \to \mathcal{Y}$ between normed spaces if $\|g(x) - g(x')\| \leq \phi(\|x - x'\|)$ for every $x, x' \in \mathcal{X}$. The modulus of continuity generalizes Lipschitz and Hölder continuity and will ultimately allow us to combine bounds on the truncation error and continuous-time optimization error (suboptimality gap) into a discrete-time suboptimality gap. With this, we assume the following ingredients in addition to the general setup of Problem 1:

1) The scheme (3) is convergent and has a known accuracy upper bound $\max_k \|\mathcal{T}_k\| \leq K(T - t_0)\omega(h)$, for some functions $K(\cdot), \alpha(\cdot)$ of class $\mathcal{K}$.
2) $V \circ G$ is continuous with modulus of continuity ϕ.

With the ingredients above, we have:

$$\begin{aligned} V(x_N) &= \big(V(x_N) - V(x(T))\big) + V(x(T)) && \text{(9a)} \\ &\leq \phi(\|X_N - X(T)\|) + V(x(T)) && \text{(9b)} \\ &\leq \phi\left(K(T - t_0)\omega(h)\right) + \alpha(V(x_0), T - t_0) && \text{(9c)} \end{aligned}$$

for all $h > 0$ and $\Delta := T - t_0$. Therefore, we can guarantee ε-optimality, i.e. $V(x_N) \leq \varepsilon$, in $N \leq \lceil \frac{\Delta}{h} \rceil$ iterations, with Δ, h chosen such that $\phi\left(K(\Delta)\omega(h)\right) + \alpha(V(x_0), \Delta) \leq \varepsilon$. In particular, we can choose $\Delta = \Delta(\varepsilon)$ such that $\alpha(V(x_0), \Delta) = \frac{\varepsilon}{2}$ and $h = h(\varepsilon)$ such that $\phi\left(K(\Delta(\varepsilon))\omega(h)\right) = \frac{\varepsilon}{2}$, thus giving us an explicit iteration complexity rate $N = \mathcal{O}\left(\frac{\Delta(\varepsilon)}{h(\varepsilon)}\right)$.

To more explicitly illustrate the approach above, let us focus on $V(x) = f(x) - f^\star$ and let us Taylor-based accuracy bounds, of the form $\mathcal{O}(h^\nu)$. Further, assuming that ∇f is bounded and F and G are sufficiently regular, we can guarantee a modulus of continuity $\phi(h) = Ch$.

Assumption 4 (ODE regularity)**.** The function $F(t, X)$ is piecewise continuous in t and uniformly Lipschitz continuous, as well as $\nu \geq 2$ times continuously differentiable, in X. Further, G is Lipschitz continuous.

Theorem 1. *Suppose that Assumptions 3,4 hold. Further, suppose that f has a bounded gradient. Consider the system* (1) *and suppose that its equilibria set is contained in $G^{-1}(\mathcal{X}^\star)$. Consider a convergent 1-step scheme*(3) *($r = 1$) of order $\mathcal{O}(h^\nu)$ that is applied to* (3)*. Let $x_k := G(X_k)$ and consider the smallest number N of steps required for $f(x_N) - f^\star \leq \varepsilon$ to hold. The following assertions hold:*

- *If* (1) *is globally asymptotically stable and such that $f(x(t)) - f^\star = \mathcal{O}((t - t_0)^{-\delta})$, then $N = \mathcal{O}(\varepsilon^{-\left(\frac{1}{\delta} + \frac{1}{\nu}\left(1 + \frac{1}{\delta}\right)\right)})$;*
- *If* (1) *is globally exponentially stable, then $N = \mathcal{O}\left(\log\left(\frac{1}{\varepsilon}\right)\varepsilon^{-\frac{1}{\nu}}\right)$;*
- *If* (1) *is globally finite-time stable, then $N = \mathcal{O}(\varepsilon^{-\frac{1}{\nu}})$,*

where the hidden constants in $\mathcal{O}(\cdot)$ depend only on $\|\nabla f\|_\infty, f(x_0) - f^\star$,$\nu$, and the scheme (3)..

Proof. Without loss of generality, we can assume $t_0 = 0$. We proceed as described earlier in the current subsection, with $V(x) = f(x) - f^\star$. Let L_G denote the Lipschitz constant of G. Then, $V \circ G$ has modulus of continuity $\phi(h) = \|\nabla f\|_\infty L_G h$. Indeed, $|(V \circ G)(X) - (V \circ G)(X')| = |\left(f(G(X)) - f^\star\right) - \left(f(G(X')) - f^\star\right)| = |f(G(X)) - f(G(X'))| \leq \|\nabla f\|_\infty \|G(X) - G(X')\| \leq \|\nabla f\|_\infty L_G \|X - X'\| = \phi(\|X - X'\|)$. Therefore, $V(x_N) \leq \phi(\|x_N - x(T)\|) + V(x(T))$. Let us consider the 3 cases in the theorem statement in their original order:

- Invoking [25, Thm. 2.1, 4.3], we know that $V(x_N) \leq C_1 T h^\nu + C_2 T^{-\delta}$ for some $C_1, C_2 > 0$ that do not depend on h or T. Let $T = T(\varepsilon)$ such that $C_2 T^{-\delta} = \varepsilon$, i.e. $T = C_2^{\frac{1}{\delta}} \varepsilon^{-\frac{1}{\delta}}$. Now let $h = h(\varepsilon)$ such that $C_1 T h^v = \varepsilon$, i.e. $h = (C_1 C_2^{\frac{1}{\delta}})^{\frac{1}{\nu}} \varepsilon^{\left(1 + \frac{1}{\delta}\right)\frac{1}{\nu}}$. Combining T, h into $N \leq \lceil T/h \rceil$ thus leads to $N \leq 1 + C_2^{\frac{1}{\delta}} C_2^{\frac{1}{\delta}\left(1 + \frac{1}{\nu}\right)} \varepsilon^{-\left[\frac{1}{\delta}\left(1 + \frac{1}{\nu}\right) + \frac{1}{\nu}\right]}$.
- Invoking [25, Thm. 2.1, 4.2], we know that $V(x_N) \leq C_1 h^\nu + C_2 e^{-C_3 T}$ for some $C_1, C_2, C_3 > 0$. Solving $C_2 e^{-C_3 T} = \varepsilon$ and $C_1 h^v = \varepsilon$, we find $N \leq 1 + \frac{C_1^{\frac{1}{\nu}}}{C_3} \log\left(\frac{C_2}{\varepsilon}\right) \varepsilon^{-\frac{1}{\nu}}$.
- Due to finite-time stability, there exists some $T(x_0) > 0$ large enough such that $x(T(x_0)) = x^\star$, independent of ε. Therefore, choosing $T = T(x_0)$, we have $V(x_N) \leq C\|x_N - x(T)\| \leq Ch^\nu$ for some $C > 0$. Choosing $h = C^{-\frac{1}{\nu}} \varepsilon^{\frac{1}{\nu}}$ therefore leads to $N \leq 1 + C^{-\frac{1}{\nu}} T(x_0) \varepsilon^{-\frac{1}{\nu}}$. ∎

Let us briefly discuss the oracle complexity implications of the previous result. To do this, suppose that the system (1) and scheme (3) satisfy Assumptions 1,2. With this, we first note that, for implicit schemes, the number of queries to the gradient ∇f heavily depends on the nature of the implicit equation to be solved. This will be the subject of future research, but for now let us specifically discuss the case of explicit Runge-Kutta schemes. Indeed, as discussed in Sec. III-A, we can construct explicit Runge-Kutta schemes of any desired order $\mathcal{O}(h^\nu)$, using at most $s = \lceil (3/8)\nu^2 \rceil$ stages. Since each iteration of an explicit s-stage Runge-Kutta scheme requires up to $\frac{s(s+1)}{2} \leq \lceil \frac{9}{128}\nu^4 \rceil N$ gradient evaluations, we can conclude that the total number of gradient evaluation required to reach ε-optimality is proportional to the number N of iterations (time steps) in the Runge-Kutta scheme. Therefore, a subexponential rate in continuous time of the form $T - t_0 = \mathcal{O}(\varepsilon^{-\frac{1}{\delta}})$ can be nearly preserved for large ν, since $\varepsilon^{-\left(\frac{1}{\delta} + \frac{1}{\nu}\left(1 + \frac{1}{\delta}\right)\right)} \approx \varepsilon^{-\frac{1}{\delta}}$. That said, the ratio between the number of gradient evaluations and N grows at a rate $\mathcal{O}(\nu^4)$, and thus using schemes of excessively high order is unlikely to be justified in practice. Instead, relatively low-order schemes (typically $\nu \leq 4$ and occasionally up to $\nu = 8$) tend to perform just as well, if not better, particularly when paired with a sensible stepsize adaptation strategy.

Example 7. Consider Nesterov's ODE (2) for convex f, so as to ensure $f(x(t)) - f^\star = \mathcal{O}(1/t^2)$. This way, assuming that f is ν times continuously differentiable and has bounded Lipschitz continuous gradient, it follows that we can discretize it using suitable scheme of order $\mathcal{O}(h^\nu)$, leading to the rate $\mathcal{O}(\varepsilon^{-\frac{1}{2}\frac{\nu+3}{\nu}})$, which is slightly worse than the corresponding rate $\mathcal{O}(\varepsilon^{-\frac{1}{2}\frac{\nu+1}{\nu}})$ derived by [17]. However, if we discretize the modified ODE

$$\ddot{x} + \frac{p+1}{t}\dot{x} + p^2 t^{p-2} \nabla f(x) = 0, \tag{10}$$

which satisfies $f(x(t)) - f^\star = \mathcal{O}(1/t^p)$ for $p > 2$ (see [9]), then we obtain the rate $\mathcal{O}(\varepsilon^{-\left(\frac{1}{p} + \frac{1}{\nu}\left(1 + \frac{1}{p}\right)\right)}) = \mathcal{O}(\varepsilon^{-\frac{1}{p}\frac{p+\nu+1}{\nu}})$. While the corresponding rate $\mathcal{O}(\varepsilon^{-\frac{1}{p}\frac{\nu+1}{\nu}})$ derived by [17] (with the term $\frac{p+1}{t}$ changed to $\frac{2p+1}{t}$) is strictly better for each $p > 2$, it should be noted that taking $\nu \to \infty$ leads to $\mathcal{O}(\varepsilon^{-\frac{1}{p}})$ in both cases (with slight abuse of notation). However, and most importantly, p in our rate is entirely free, unlike in [17] where it relates to a flatness condition that scales with p and is rather restrictive for large p. With this in mind, using our approach, we can freely take $p \to \infty$ to obtain the rate $\mathcal{O}(\varepsilon^{-\frac{1}{\nu}})$ (with slight abuse of notation), nearly matching the best possible rate of $\tilde{\mathcal{O}}(\varepsilon^{-\frac{2}{3\nu+1}})$ [26] when we allow to query the first ν derivatives of f.

IV. The rescaled gradient flow

To further illustrate the findings of Sec. III-B, we now consider the following candidate continuous-time optimization algorithm, known as the q-rescaled gradient flow (RGF):

$$\dot{x} = -\frac{\nabla f(x)}{\|\nabla f(x)\|^{\frac{q-2}{q-1}}}, \tag{11}$$

where $1 < q < \infty$. Note that $\|\dot{x}(t)\| = \|\nabla f(x(t))\|^{\frac{1}{q-1}}$. Therefore, as $x(t) \to x^\star$, the role of q can be seen as leading to acceleration (compared to the gradient flow) for $q > 2$ and deceleration for $1 < q < 2$. For $q = 2$, the system simplifies to the usual gradient flow $\dot{x} = -\nabla f(x)$, whereas for $q = \infty$ it simplifies to $\dot{x} = -\frac{\nabla f(x)}{\|\nabla f(x)\|}$ (interpreted as a Filippov differential inclusion; see [27]), which was introduced in [28], [29] as the *normalized* gradient flow. The q-RGF was proposed in its current form in [9] and has also been studied in [14] and [30].

In order to establish accelerated convergence, additional regularity is required:

Assumption 5 (Gradient dominance)**.** The function f is *μ-gradient dominated* of order p, i.e.

$$\frac{p-1}{p}\|\nabla f(x)\|^{\frac{p}{p-1}} \geq \mu^{\frac{1}{p-1}}(f(x) - f^\star) \tag{12}$$

for all $x \in \mathbb{R}^n$, where $f^\star = \inf_x f(x)$, $\mu > 0$, and $p > 1$.

This terminology is borrowed and adapted from [9], [18]. We also utilized the notion of gradient dominance of order p in earlier work [14] and it has also been leveraged as a particular case of interest of the more general Kurdyka-Łojasiewicz (KL) inequality. In particular, gradient dominance is nothing more than the Łojasiewicz gradient inequality, with the particular case $p = 2$ known as the Polyak-Łojasiewicz (PL) inequality (or simply *gradient dominance*) because it was independently studied by Łojasiewicz [31] and Polyak [32] in 1963. Note that gradient dominance guarantees invexity.

Before we summarize the convergence guarantees that gradient dominance give up on the RGF, let us briefly describe when it would hold.

Proposition 1. *If f is μ-uniformly convex of order p, i.e.*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle + \frac{\mu}{p}\|x - y\|^p, \tag{13}$$

for all $x, y \in \mathbb{R}^n$, then it is μ-gradient dominated of order p.

Proof. Let us minimize the RHS of (13). Differentiating, we find that the minimum is attained at $y^\star$ such that $\mu\|x - y^\star\|^{p-2}(x - y^\star) = \nabla f(x)$. Therefore, $\|x - y^\star\| = \mu^{-\frac{1}{p-1}}\|\nabla f(x)\|^{\frac{1}{p-1}}$ and $\frac{x-y^\star}{\|x-y^\star\|} = \frac{\nabla f(x)}{\|\nabla f(x)\|}$. Subsequently, $x - y^\star = \mu^{-\frac{1}{p-1}}\nabla f(x)/\|\nabla f(x)\|^{\frac{p-2}{p-1}}$. Plugging $y = \arg\min_x f(x)$ on the LHS of (13) and $y = y^\star$ on the RHS, we find (12) after simplifying. ■

More generally, the inequality was first established to hold locally for analytic functions [31], [33]. Later, analyticity was relaxed to (possibly non-differentiable) subanalytic functions and it was proved that the inequality holds globally under convexity [34, Thm. 3.1 & 3.3]. More generally, we can characterize gradient dominance as follows:

Proposition 2. *If f is μ-gradient dominated of order p, then it satisfies the Hölderian error bound (HEB)*

$$f(x) - f^\star \geq C_{p,\mu} \cdot \|x - x^\star\|^p, \tag{14}$$

for all $x \in \mathbb{R}^n$, where $C_{p,\mu} = \frac{\mu}{p(p-1)^{p-1}}$. Reciprocally, if f is convex and (14) *holds, then f is (μ/p^p)-gradient dominated of order p.*

Remark 1. The HEB (14) is equivalent to (13) with $x = x^\star$ fixed in (13) (i.e. uniform convexity *at* $x^\star$ only [**?**]) and μ replaced by $\mu/(p-1)^{p-1}$. Therefore, HEBs are closely closely related to but weaker than uniform convexity. The particular relationship between the case $p = 2$ of (14) (known as *quadratic growth* in the literature), the PL inequality, and strong (or weakly strong) convexity was highlighted in [35], among related conditions. It is also worth noting that convexity in the above proposition could be replaced by invexity (Assumption 3), provided that (14) is replaced by $f(x) - f^\star \geq C_{p,\mu}\|\xi(x, x^\star)\|^p$, where the case $\xi(x, y) = y - x$ corresponds to convexity.

We conclude this subsection by summarizing the convergence rate of q-RGF in terms under gradient dominance order p by pooling and slightly generalizing some of the results proved in [9], [18] and [14], [30].

Proposition 3. *Suppose that f satisfies Assumption 5 and let $x(t)$ be a maximal solution to* (11) *with initial condition $x(0) = x_0$. Then, the following convergence rates are attained for $V(x) := f(x) - f^\star$:*

- *(sublinear/subexponential) If $q < p$, then*

$$V(x(t)) \leq [V(x_0)^{-\delta} + \beta t]^{-\delta} = \mathcal{O}(t^{-\delta}), \tag{15}$$

where $\delta = 1/\left(\frac{p-1}{p} \cdot \frac{q}{q-1} - 1\right) \in (0, \infty)$ and $\beta = \left(\frac{p}{p-1}\right)^{\delta+1} \mu^{\frac{q}{p(q-1)}}\delta \in (0, \infty)$;

- *(linear/exponential) If $q = p$, then*

$$V(x(t)) \leq V(x_0)\,\mathrm{e}^{-\frac{p}{p-1}\mu^{\frac{1}{p-1}}t} = \mathcal{O}(\rho^t), \tag{16}$$

with $\rho = \exp\left(-\frac{p}{p-1}\mu^{\frac{1}{p-1}}\right) \in (0, 1)$;

- *(finite time) If $q > p$, then $V(x(t)) = 0$ for every $t \geq t^\star$, with $t^\star \leq \frac{\|\nabla f(x_0)\|^{\frac{1}{\theta} - \frac{1}{\theta'}}}{C^{\frac{1}{\theta}}\left(1 - \frac{\theta}{\theta'}\right)} < \infty$, where $C = \left(\frac{p}{p-1}\right)^{\frac{p-1}{p}}\mu^{\frac{1}{p}}$, $\theta = \frac{p-1}{p}$, and $\theta' = \frac{q-1}{q}$.*

Proof (sketch)**.** Let $\mathcal{E}(t) = f(x(t)) - f^\star$. In all three cases, we will have $\dot{\mathcal{E}}(t) \leq -c\,\mathcal{E}(t)^\alpha$ with $c, \alpha > 0$. Indeed,

$$\dot{\mathcal{E}}(t) = \langle \nabla f(x(t)), \dot{x}(t)\rangle \tag{17a}$$

$$= -\|\nabla f(x(t))\|^{\frac{q}{q-1}} \tag{17b}$$

$$\leq -C^{\frac{1}{\theta'}}\mathcal{E}(t)^{\frac{\theta}{\theta'}}, \tag{17c}$$

where the last inequality follows by invoking gradient dominance. Clearly, depending on whether q is below, equal, or above p, we will have that α is above, equal, or below 1. Thus, integrating the differential inequality $\dot{\mathcal{E}}(t) \leq -c\,\mathcal{E}(t)^{\alpha}$, we find the rates described. ■

Remark 2. Rates for $V(x) = \|x - x^{\star}\|$ can be directly obtained from the previous result by exploiting the HEB from Proposition 2. Likewise, rates for $V(x) = \|\nabla f(x)\|$ can be determined under the additional assumption of L-smoothness or Hölder continuous gradient, for instance.

We conclude this section by applying the result from Sec. III-B to the rescaled gradient flow.

Theorem 2. *Suppose that f has bounded gradient and satisfies satisfies Assumption 5. Further, suppose that the RHS of* (11) *is Lipschitz continuous and ν times continuously differentiable. Consider a convergent 1-step scheme* (3) *of order $\mathcal{O}(h^{\nu})$. Then, applying the scheme to* (11) *on $[t_0, T]$, we have $f(x_N) - f^{\star} \leq \varepsilon$, with*

$$N = \begin{cases} \mathcal{O}\left(\varepsilon^{-\left[\left(\frac{p-1}{p}\cdot\frac{q}{q-1}-1\right)\left(1+\frac{1}{\nu}\right)+\frac{1}{\nu}\right]}\right), & \text{if } q < p \\ \mathcal{O}\left(\log\left(\frac{1}{\varepsilon}\right)\varepsilon^{-\frac{1}{\nu}}\right), & \text{if } q = p \\ \mathcal{O}(\varepsilon^{-\frac{1}{\nu}}), & \text{if } q > p, \end{cases} \tag{18}$$

where the multiplicative constants hidden in $\mathcal{O}(\cdot)$ depend only on $\|\nabla f\|_{\infty}, f(x_0) - f^{\star}, \mu, p, q, \nu$, and the scheme (3).

V. Conclusion and Future Work

Traditionally, optimization algorithms are designed as iterative schemes. Such schemes have a natural temporal interpretation: consecutive iterates may be interpreted as evolving over discrete time time steps. This simple realization has motivated researchers to borrow tools from the fields of dynamical systems and control theory. While this approach has been gaining traction in the last few years, the modeling tools used in systems and control are largely based around *continuous-time* representations of a system's evolution over time, typically via ordinary differential equations.

Nonetheless, there has been great success recently in studying continuous-time analogues of optimization algorithms, enabling new theoretical analysis and design approaches. However, there is also clearly a contrast between continuous time being ubiquitous in systems and control, but discrete time being practically a necessity for optimization algorithms in practice on account of the digital nature of modern computers. For these reasons, we believe that it is worthwhile to search for a systematic theory of discretization fine-tuned for the purpose of optimization.

To this end, in this paper we focus on the fundamental question of how to discretize such a continuous-time analogue of an optimization algorithm (with a focus on gradient-based ones) in such a way so as preserve some of its optimality properties that were originally present in continuous time. In particular, by considering off-the-shelf ODE solvers, we can combine their built-in accuracy bounds with the suboptimality bounds known for the continuous-time optimization algorithm to therefore bound the discretization's suboptimality gap.

With this approach, we show that for sufficiently stable systems with convergence rates of three major types (subexponential, exponential, and finite-time convergent), we can nearly match their original rates upon discretization by merely using ODE solvers of modestly high order. In particular, we show that an ε-optimality rates of the form $\mathcal{O}(1/\varepsilon^{1/\delta}), \mathcal{O}(\log(1/\varepsilon)), \mathcal{O}(1)$ in continuous time can be readily converted into corresponding rates $\mathcal{O}(1/\varepsilon^{\frac{1}{\delta}+\frac{1}{\nu}\left(1+\frac{1}{\delta}\right)}), \mathcal{O}(\log(1/\varepsilon)/\varepsilon^{1/\nu}), \mathcal{O}(1/\varepsilon^{1/\nu})$, respectively, upon discretization with a solver of accuracy order $\mathcal{O}(h^{\nu})$. Therefore, for modestly large ν, the rates will approximately match.

As a particular case study, we first considered the modified version (10) of Nesterov's ODE 2 and found a novel rate $\mathcal{O}(1/\varepsilon^{\frac{1}{p}\frac{p+\nu+1}{\nu}})$. When contrasted with the rate $\mathcal{O}(1/\varepsilon^{\frac{1}{p_{\max}}\frac{\nu+1}{\nu}})$ derived in [17], where $p_{\max}$ originates from a flatness condition, we can see that our rate can outperforms it for certain combinations of p and ν. Further, we can see that, when taking $p \to \infty$, our rate approaches $\mathcal{O}(1/\varepsilon^{\frac{1}{\nu}})$, which is remarkably close to the known upper bound $\mathcal{O}(1/\varepsilon^{\frac{1}{\nu-1}})$ and lower bound $\Omega(1/\varepsilon^{\frac{2}{3\nu+1}})$ attainable with oracle access to the first ν derivatives of the cost function. As a second case study, we consider the rescaled gradient flow (11) and provide novel rates summarized in Theorem 2. In particular, for the cases where the assumed gradient dominance is of order $p = \infty$ and $p = q$, we recover the corresponding rates in [9], [18] under weaker conditions, when using high-order solvers.

The novel rates that we obtained in the two case studies proved to be competitive with the existing rates in the literature, even though these required stronger assumptions and a highly tailored analysis. While this is encouraging, we find that accuracy orders of the form $\mathcal{O}(h^{\nu})$ act as a bottleneck to establish true linear convergence (i.e., without $\nu \to \infty$, which is impractical). In future work, we aim to fill this gap by considering a more sophisticated approach based around nonlinear absolute stability of ODE solvers. Further, we will explore multiderivative schemes. Lastly, we seek to extended our methodology for stochastic, online, robust, and distributed optimization.

VI. ACKNOWLEDGMENTS

This research is partially supported by AFOSR under grant FA9550-19-1-0265 (Assured Autonomy in Contested Environments).

References

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

[2] G. França, J. Sulam, D. Robinson, and R. Vidal, "Conformal symplectic and relativistic optimization," *Neural Information Processing Systems (NeurIPS 2020)*, December 2020.

[3] M. Muehlebach and M. Jordan, "A dynamical systems perspective on Nesterov acceleration," in *Proceedings of the 36th International Conference on Machine Learning*. PMLR, June 2019, pp. 4656–4662.

[4] H. Luo and L. Chen, "From differential equation solvers to accelerated first-order methods for convex optimization," *Mathematical Programming*, 2021.

[5] A. K. Zghier, “The use of differential equations in optimization,” Ph.D. dissertation, Loughborough University, 1981.

[6] U. Helmke and J. B. Moore, *Optimization and Dynamical Systems*. Springer-Verlag, 1994.

[7] 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*, December 2011, pp. 3800–3805.

[8] W. Su, S. Boyd, and E. J. Candès, “A differential equation for modeling nesterov’s accelerated gradient method: Theory and insights,” *Journal of Machine Learning Research*, vol. 17, no. 1, p. 5312–5354, January 2016.

[9] A. Wibisono, A. C. Wilson, and M. I. Jordan, “A variational perspective on accelerated methods in optimization,” *Proceedings of the National Academy of Sciences*, vol. 113, no. 47, pp. E7351–E7358, 2016.

[10] 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)*, May 2017, pp. 5269–5275.

[11] G. França, D. Robinson, and R. Vidal, “ADMM and accelerated ADMM as continuous dynamical systems,” July 2018.

[12] H. Attouch, Z. Chbani, J. Fadili, and H. Riahi, “Convergence of iterates for first-order optimization algorithms with inertia and hessian driven damping,” *Optimization*, vol. 0, no. 0, pp. 1–40, 2021.

[13] H. Attouch, A. Balhag, Z. Chbani, and H. Riahi, “Fast convex optimization via inertial dynamics combining viscous and hessian-driven damping with time rescaling,” *Evolution Equations Control Theory*, vol. 11, no. 2, pp. 487–514, 2022.

[14] O. Romero and M. Benosman, “Finite-time convergence in continuous-time optimization,” in *the 37th International Conference on Machine Learning*, vol. 119. PMLR, July 2020, pp. 8200–8209.

[15] K. Garg and D. Panagou, “Fixed-time stable gradient-flow schemes: Applications to continuous-time optimization,” *Transactions on Automatic Control*, vol. 66, no. 5, pp. 2001–2015, 2020.

[16] H. K. Khalil, *Nonlinear Systems*. Englewood Cliffs, New Jersey: Prentice-Hall, 2001.

[17] J. Zhang, A. Mokhtari, S. Sra, , and A. Jadbabaie, “Direct runge-kutta discretization achieves acceleration,” in *the 32rd Conference on Neural Information Processing Systems (NeurIPS 2018))*, December 2018.

[18] A. Wilson, L. Mackey, and A. Wibisono, “Accelerating rescaled gradient descent: Fast optimization of smooth functions,” in *Advances in Neural Information Processing Systems*, December 2019.

[19] J. Butcher, “General linear methods,” *Computers Mathematics with Applications*, vol. 31, no. 4, pp. 105–112, 1996, selected Topics in Numerical Methods.

[20] J. C. Butcher, “On the convergence of numerical solutions to ordinary differential equations,” *Mathematics of Computation*, vol. 20, no. 93, pp. 1–10, 1966. [Online]. Available: http://www.jstor.org/stable/2004263

[21] J. Butcher, *Numerical Methods for Ordinary Differential Equations*, 2nd ed. John Wiley & Sons, 2016.

[22] A. M. Stuart and A. R. Humphries, *Dynamical systems and numerical analysis*, 1st ed. Cambridge University Press, November 1998.

[23] A. Orvieto and A. Lucchi, “Shadowing properties of optimization algorithms,” in *Neural Information Processing Systems*, December 2019.

[24] A. Iserles and G. Söderlind, “Global bounds on numerical error for ordinary differential equations,” *Journal of Complexity*, vol. 9, no. 1, pp. 97–112, 1993.

[25] D. Viswanath, “Global errors of numerical ODE solvers and Lyapunov’s theory of stability,” *IMA Journal of Numerical Analysis*, vol. 21, no. 1, pp. 387–406, 01 2001.

[26] Y. E. Nesterov, “Implementable tensor methods in unconstrained convex optimization,” *Math. Program.*, vol. 186, no. 1, pp. 157–183, 2021. [Online]. Available: https://doi.org/10.1007/s10107-019-01449-1

[27] J. Cortés, “Discontinuous dynamical systems,” *IEEE Control Systems Magazine*, vol. 28, no. 3, pp. 36–73, June 2008.

[28] J. Cortés and F. Bullo, “Coordination and geometric optimization via distributed dynamical systems,” *SIAM Journal on Control and Optimization*, vol. 44, no. 5, pp. 1543–1574, October 2005.

[29] J. Cortés, “Finite-time convergent gradient flows with applications to network consensus,” *Automatica*, vol. 42, no. 11, pp. 1993–2000, November 2006.

[30] K. Garg and D. Panagou, “Fixed-time stable gradient flows: Applications to continuous-time optimization,” *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2002–2015, 2021.

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

[32] B. Polyak, “Gradient methods for the minimisation of functionals (in Russian),” *USSR Computational Mathematics and Mathematical Physics*, vol. 3, pp. 864–878, December 1963.

[33] S. Łojasiewicz, *Ensembles semi-analytiques*. Centre de Physique Theorique de l’Ecole Polytechnique, 1965. [Online]. Available: https://perso.univ-rennes1.fr/michel.coste/Lojasiewicz.pdf

[34] J. Bolte, A. Daniilidis, and A. Lewis, “The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems,” *Society for Industrial and Applied Mathematics*, vol. 17, pp. 1205–1223, January 2007.

[35] H. Karimi, J. Nutini, , and M. 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*. Springer, 2016, pp. 795–811.

Appendix

A. Proof of Proposition 2

First, note that (12) can be restated in the Łojasiewicz gradient inequality form

$$\|\nabla f(x)\| \geq C(f(x) - f^\star)^\theta, \tag{19}$$

with $C = \left(\frac{p}{p-1}\right)^{\frac{p-1}{p}} \mu^{\frac{1}{p}} > 0$ and $\theta = \frac{p-1}{p} \in (0,1)$. Let $g(x) = (f(x) - f^\star)^{1-\theta}$ and let $x(t)$ denote the maximal solution of $\dot{x} = -\nabla g(x)$ such that $x(t) \neq x^\star$, with initial state $x(0) = x_0 \neq x^\star$. We have $\nabla g(x) = (1-\theta)\frac{\nabla f(x)}{(f(x)-f^\star)^\theta}$, and thus $\|\nabla g(x)\| \geq C(1-\theta)$. Therefore,

$$\begin{aligned}
\frac{d}{dt} g(x(t)) &= \langle \nabla g(x(t)), \dot{x}(t) \rangle && \text{(20a)}\\
&= -\|\nabla g(x(t))\|^2 && \text{(20b)}\\
&\leq -C^2(1-\theta)^2, && \text{(20c)}
\end{aligned}$$

and thus, integrating and noting that $g(x_0) > 0$, we find that $g(x(t)) \to 0$ as $t \to T$ for some finite-valued $T = T(x_0) > 0$. Therefore, $x(T) := \lim_{t\to\infty} x(t)$ is actually $x(T) = x^\star$. Further, we have

$$\begin{aligned}
g(x_0) &= g(x_0) - g(x(T)) && \text{(21a)}\\
&= -\int_0^T \frac{d}{dt} g(x(t)) dt && \text{(21b)}\\
&= \int_0^T \|\nabla g(x(t))\|^2 dt && \text{(21c)}\\
&\geq C(1-\theta) \int_0^T \|\nabla g(x(t))\| dt && \text{(21d)}\\
&= C(1-\theta) \int_0^T \|\dot{x}(t)\| dt && \text{(21e)}\\
&\geq C(1-\theta)\|x(T) - x_0\| && \text{(21f)}\\
&= C(1-\theta)\|x_0 - x^\star\|. && \text{(21g)}
\end{aligned}$$

Rearranging terms, we find $f(x) - f^\star \geq C^{\frac{1}{1-\theta}}(1-\theta)^{\frac{1}{1-\theta}}\|x - x^\star\|^{\frac{1}{1-\theta}}$. Substituting C and θ in terms of μ and p and simplifying, the proof is complete. ■