Formatting Instructions For NeurIPS 2022

Xiaoxuan Yu
College of Chemistry and Molecular Engineering
Peking University
Beijing, China
xiaoxuan_yu@pku.edu.cn

Yihang Xia
School of Mathematical Sciences
Peking University
Beijing, China
xyh-mathematics@pku.edu.cn

Abstract

The abstract paragraph should be indented ½ inch (3 picas) on both the left- and right-hand margins. Use 10 point type, with a vertical spacing (leading) of 11 points. The word **Abstract** must be centered, bold, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

1 Problem setup and backgrounds

The article studies gradient-based optimization methods obtained by directly discretizinga second-order ordinary differential equation (ODE) related to the continuous limit of Nesterov's accelerated gradient method. It introduces some conditions under which the sequence of iterates generated by discretizing the proposed second-order ODE converges to the optimal solution at a certain rate.

Firstly let's focus the essential target problem:

$$\min_{x\in\mathbb{R}^d} f(x), \tag{1}$$

where f is convex and sufficiently smooth. For solving (1), there are several methods.

- Classical method: gradient decent, which displays a sub-optimal convergence rate of $\mathcal{O}(N^{-1})$.
- Nesterov's seminal accelerated gradient method, matches the oracle lower bound of $\mathcal{O}(N^{-2})$.

Several articles have pursued approaches to NAG (and accelerated methods in general) via a continuous-time perspective. However, they fail to provide a general discretization procedure that generates provably convergent accelerated methods. This article takes Runge-Kutta integrater as tool and introduces a second-order ODE that generates an accelerated first-order method for smooth functions if using Runge-Kutta method.

Place for introduction to additional related work if necessary.

2 Main results

To build up a iterative method, letting x_0 be the initial point, firstly we consider the sublevel set

$$\mathcal{S} := \{x \in \mathbb{R}^d | f(x) \leq \exp(1)(f(x_0) - f(x^*) + \|x_0 - x^*\|^2) + 1\}, \tag{2}$$

where x^* is the minimum of (1). The introduction of set $\mathcal{S}$ actually gives a restriction of the sequence of iterates obtained from discretizing a suitable ODE (would be proved later). Denote subset

$$\mathcal{A} := \{x \in \mathbb{R}^d | \exists x' \in \mathcal{S}, \|x - x'\| \leq 1\}, \tag{3}$$

Then all assumptions we may require can be considered just in $\mathcal{A}$.

Preprint. Under review.

Assumption 1. *There exists an integer $p \geq 2$ and a positive constant L such that for any point $x \in \mathcal{A}$, and for all indices $i \in \{1, \dots, p-1\}$, we have the lower-bound*

$$f(x) - f(x^*) \geq \frac{1}{L} \|\nabla^{(i)} f(x)\|^{\frac{p}{p-i}}, \tag{4}$$

where x^* minimizes f and $\|\nabla^{(i)} f(x)\|$ denotes the operator norm of the tensor $\nabla^{(i)} f(x)$.

Assumption 2. *There exists an integer $s \geq p$ and a constant $M \geq 0$, such that $f(x)$ is order $(s+2)$ differentiable. Furthermore, for any $x \in \mathcal{A}$, the following operator norm bounds hold:*

$$\|\nabla^{(i)} f(x)\| \leq M, \text{ for } i = p, p+1, \dots, s, s+1, s+2. \tag{5}$$

When the sublevel sets of f are compact and hence the set $\mathcal{A}$ is also compact; as a result, the bound (5) on high order derivatives is implied by continuity.

2.1 Runge-Kutta integrators

The explicit Runge-Kutta integrators used in the article appears in the form below.

Definition 1. *Given a dynamical system $\dot{y} = F(y)$, let the current point be y_0 and the step size be h. An explicit S stage Runge-Kutta method generates the next step via the following update:*

$$g_i = y_0 + h \sum_{j=1}^{i-1} a_{ij} F(g_j), \quad \Phi_h(y_0) = y_0 + h \sum_{i=1}^{S} b_i F(g_i), \tag{6}$$

where a_{ij} and b_i are suitable coefficients defined by the integrator; $\Phi_h(y_0)$ is the estimation of the state after time step h, while g_i (for $i = 1, \dots, S$) are a few neighboring points where the gradient information $F(g_i)$ is evaluated.

In general, Runge-Kutta methods offer a powerful class of numerical integrators, and with the knowledge of its concergence behaviour when discretizing for solutions, the article gets to use it to discretize ODE with controlment of its convergence rates.

2.2 Formal work and inspiration

Then focus on the NAG method that is defined according to the updates

$$x_k = y_{k-1} - h\nabla f(y_{k-1}), \quad y_k = x_k + \frac{k-1}{k+2}(x_k - x_{k-1}). \tag{7}$$

Su, Boyd, and Candès [3] showed that the iteration (7) in the limit is equivalent to the following ODE

$$\ddot{x}(t) + \frac{3}{t}\dot{x}(t) + \nabla f(x(t)) = 0, \quad \text{where } \dot{x} = \frac{dx}{dt} \tag{8}$$

when one drives the step size h to zero. It can be further shown that in the continuous domain the function value $f(x(t))$ decreases at the rate of $\mathcal{O}(1/t2)$ along the trajectories of the ODE. This convergence rate can be accelerated to an arbitrary rate in continuous time via time dilation as in [Wibisono et al., 2016]. In particular, the solution to

$$\ddot{x}(t) + \frac{p+1}{t}\dot{x}(t) + p^2 t^{p-2} \nabla f(x(t)) = 0 \tag{9}$$

has a convergence rate $\mathcal{O}(1/t^p)$. When $p > 2$, Wibisono et al. [2016] proposed rate matching algorithms via utilizing higher order derivatives. However, this article focuses purely on first-order methods and study the stability of discretizing the ODE directly when $p \geq 2$.

According to some related work, deriving the ODE from the algorithm is now a solved problem. Nevertheless, to derive the update of NAG or any other accelerated method by directly discretizing an ODE is not. Some work points out that explicit Euler discretization of the ODE in (8) may not lead to a stable algorithm. Betancourt, Jordan, and Wilson [1] observed empirically that Verlet integration is stable and suggested that the stability relates to the symplectic property of the Verlet integration, but for dissipative systems such as (9), this doesn't hold. This article offers a different approach: it augments the state with time in (9), and focuses the following ODE

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

This actually turns the non-autonomous dynamical system into an autonomous one.

2.3 Main conclusion

The ODE in (10) can also be written as the dynamical system

$$\dot{y} = F(y) = \begin{bmatrix} -\frac{2p+1}{t}v - p^2 t^{p-2}\nabla f(x) \\ v \\ 1 \end{bmatrix}, \quad \text{where } y = [v; x; t]. \tag{11}$$

Algorithm 1: Input(f, x_0, p, L, M, s, N) ▷ Constants p, L, M are the same as in Assumptions
1. Set the initial state $y_0 = [\overrightarrow{0}; x_0; 1] \in \mathbb{R}^{2d+1}$
2. Set step size $h = C/N^{(1/s+1)}$ ▷ C is determined by p, L, M, s, x_0
3. $x_N \leftarrow$ Oreder-s-Runge-Kutta-Integrater(F, y_0, N, h) ▷ F is defined in equation (11)
4. **return** x_N

Since the article has augmented the state with time to obtain an autonomous system, it can be readily solved numerically with a Runge-Kutta integrator as in **Algorithm 1**.

Theorem 1. *Consider the second-order ODE in* (10)*. Suppose that the function f is convex and Assumptions 1 and 2 are satisfied. Further, let s be the order of the Runge-Kutta integrator used in **Algorithm 1**, N be the total number of iterations, and x_0 be the initial point. Also, let $\mathcal{E}_0 := f(x_0) - f(x^*) + \|x_0 - x^*\|^2 + 1$. Then, there exists a constant C_1 such that if we set the step size as $h = C_1 N^{-1/(s+1)}(L + M + 1)^{-1}\mathcal{E}_0^{-1}$, the iterate x_N generated after running **Algorithm 1** for N iterations satisfies the inequality*

$$f(x_N) - f(x^*) \leq C_2 \mathcal{E}_0 \left[\frac{(L + M + 1)\mathcal{E}_0}{N^{\frac{s}{s+1}}}\right]^p = \mathcal{O}(N^{-p\frac{s}{s+1}}), \tag{12}$$

where the constants C1 and C2 only depend on s, p, and the Runge-Kutta integrator. Since each iteration consumes S gradient, $f(x_N) - f(x^*)$ will converge as $\mathcal{O}(S^{\frac{ps}{s+1}} N^{-\frac{ps}{s+1}})$ with respect to the number of gradient evaluations. Note that for commonly used Runge-Kutta integrators, $S \leq 8$.

3 Numerical Experiments

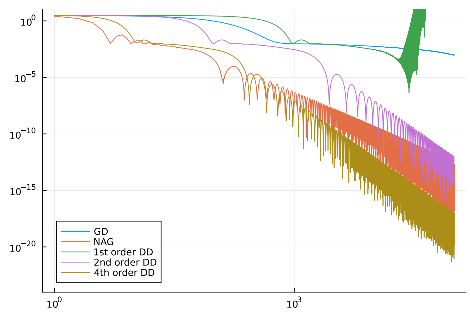
In this section, we implement the algorithms in the original article with `Julia` and its package `DifferentialEquations.jl` [2]. By comparing ODE direct discretizating (DD) methods described in the article against gradient descent (GD) and Nesterov's accelerated gradient (NAG) methods, we con verify the main results in the theoretical part. The code of these experiments can be found here: `https://github.com/xiaoxuan-yu/Direct-Runge-Kutta-Discretization-Achieves-Acceleration-PKU`.

Inspired by the numerical results by Wilson, Mackey, and Wibisono [4], we generate normal distributed separable dataset and fit a linear model $Ax = b$. Then, we minimize three different kinds of loss functions:

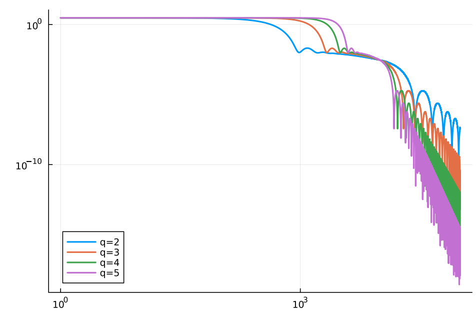
$$\begin{aligned} f_1(x) &= \|Ax - b\|_2^2 \\ f_2(x) &= \sum_i \log(1 + e^{-w_i^{\mathrm{T}} x y_i}) \\ f_3(x) &= \frac{1}{4}\|Ax - b\|_4^4 \end{aligned} \tag{13}$$

where $f_1(\cdot), f_2(\cdot), f_3(\cdot)$ are L_2 loss, logistic loss and L_4 loss, respectively. For each test case and optimization algorithm, we empirically select the learning rate as the largest step length among $\{10^{-k} | k \in \mathbb{Z}\}$ that the method remains stable during the optimization process. Main results are shown in Figure 1 where all figures are on log-log scale.

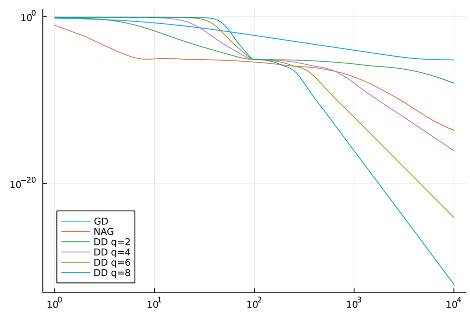
First, we explore the optimization path of a quadratic function, the L_2 loss, w.r.t. iteration. In particular, we labeled half of the generated data by 0 and the rest by 1. In Figure 1a, the ODE is discretized for $p = 2$ with different Runge-Kutta integrators with $s \in \{1, 2, 4\}$ and compared against GD and NAG algorithm. We can find that except the integrator with $s = 1$ can not converge due to the unstability of the differential format itself, the DD methods shows superiority over GD. By using higher order iterator, the local acceleration is achieved and 4th order DD even converges faster than NAG (although for each iteration, it is obviously more costly than NAG). In Figure 1b, we explore



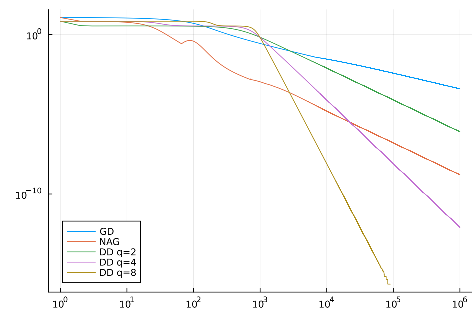
(a)



(b)



(c)



(d)

Figure 1: Experimental results comparing DD with GD and NAG. (a) Convergence path of GD, NAG and DD with different Runge-Kutta integrators of degree $s = 1, 2, 4$ on L_2 loss. (b) The optimization of L_2 loss by DD with different choices of q values with 4-th order Runge-Kutta integrator RK4. (c) Minimization of L_4 loss by GD, NAG and DD with different q vaues with a 2-nd order Runge-Kutta integrator. (d) Minimization of logistic loss by GD, NAG and DD with different q vaues with a 4-th order Runge-Kutta integrator.

the effect of q is the ODE. Since in the article p keeps the same as the one in the assumption, thus we denotes q the true parameter used in the ODE as below

$$\ddot{x}(t) + \frac{2q+1}{t}\dot{x}(t) + q^2 t^{q-2} \nabla f(x(t)) = 0.$$

We optimize the same L_2 loss with different values of q. By selecting smaller learning rates and increasing the numerical precision by using longer floats, the phenomenon that DD method diverges when $q > 2$ is not observed. Instead, we found that for $q \in \{2, 3, 4, 5\}$, larger q will give out faster convergence.

Then the minimization of L_4 loss (Figure 1c) and logistic loss (Figure 1d) is studied. We use 2-nd order Runge-Kutta integrator SSPRK22 for logistic loss optimization and 4-th order Runge-Kutta integrator RK4 for L_4 loss. As shown in Figure 1c and 1d, the loss decrease faster for larger q, as we can observed in above experiment about L_2 loss.

4 Discussion

4.1 Intuitive knowledge

Roughly speaking, this article allows for the design of optimization methods via direct discretization using Runge-Kutta integrators. However, the two assumptions required would be essential. **Assumption 1** quantifies the local flatness of convex functions in a way, and it actually contradicts our normal impression that gradient descent converges fast when the objective is not flat. This innovative discovery may inspire people to hold a more modern opinion towards the connection between convergence and local flatness. Also, the article claims that with careful analysis, discretizing ODE can preserve some of its trajectories properties. As a result, making further research on continuous ODE or appling the KR method to more general ODE cases can be valuable.

4.2 Potential research directions

To make further steps, there are quite some choices to take. The article uses conditions of higher-order differentiability to finally achieve an algorithm involving only first-order differential. We can see if allowing second and higher-order differential in the final algorithm will make things different, though in that case NAG method would be useless so we have to find another acceleration method to start with. Furthermore, as discussed above, the influence of local flatness to the convergence behaviour in discretized integraters is worth digging. How does the process of integration approaching actually work? What's the instinctive impact of local differentials and higher-order differentials? With techniques we know, some new results might be discovered.

To make a bold move, adding some random part to the conditions might leads to some interesting facts.

If someone comes up with more technical idea or just something creative, we can add here.

References

[1] Michael Betancourt, Michael I. Jordan, and Ashia C. Wilson. *On Symplectic Optimization*. 2018. DOI: 10.48550/ARXIV.1802.03653. URL: https://arxiv.org/abs/1802.03653.

[2] Christopher Rackauckas and Qing Nie. "Differentialequations.jl–a performant and feature-rich ecosystem for solving differential equations in julia". In: *Journal of Open Research Software* 5.1 (2017).

[3] Weijie Su, Stephen Boyd, and Emmanuel J. Candès. "A Differential Equation for Modeling Nesterov's Accelerated Gradient Method: Theory and Insights". In: *Journal of Machine Learning Research* 17.153 (2016), pp. 1–43. URL: http://jmlr.org/papers/v17/15-084.html.

[4] Ashia C Wilson, Lester Mackey, and Andre Wibisono. "Accelerating Rescaled Gradient Descent: Fast Optimization of Smooth Functions". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/7a2b33c672ce223b2aa5789171ddde2f-Paper.pdf.