Comparative Analysis of Accelerated Gradient Algorithms for Convex Optimization: High and Super Resolution ODE Approach

Samir Adly, Hedy Attouch, Jalal M. Fadili

Comparative Analysis of Accelerated Gradient Algorithms for Convex Optimization: High and Super Resolution ODE Approach.

Samir Adly* Hedy Attouch† Jalal Fadili‡

May 21, 2023

Abstract

We investigate convex differentiable optimization and explore the temporal discretization of damped inertial dynamics driven by the gradient of the objective function. This leads to three accelerated gradient algorithms: Nesterov Accelerated Gradient (NAG), Ravine Accelerated Gradient (RAG), and (IGAHD). Attouch, Chbani, Fadili, and Riahi introduced (IGAHD) by discretizing inertial dynamics with Hessian-driven damping to attenuate inherent oscillations in inertial methods. By analyzing the high-resolution ODEs of order $p = 0, 1, 2$ for these algorithms, we gain insights into their similarities and differences. All three algorithms share the same low-resolution ODE of order 0, which is the dynamic proposed by Su, Boyd, and Candès as a continuous surrogate for (NAG). To differentiate Nesterov from Ravine, we refine the comparison and demonstrate distinct high-resolution ODEs of order 2 in h (termed super-resolution). The corresponding Taylor expansions in h reveal matching terms of order 1 but differing terms of order 2. To the best of our knowledge, this result is completely new and emphasizes the need to avoid confusion between the Ravine and Nesterov methods in the literature. We present numerical experiments to illustrate our theoretical results. Performance profiles, measuring the number of iterations, indicate that (IGAHD) outperforms both (NAG) and (RAG) methods. (RAG) exhibits a slight advantage over (NAG) in terms of the average number of iterations. When considering CPU-time, both (RAG) and (NAG) outperform (IGAHD). All three algorithms exhibit similar behavior when evaluating based on gradient norms.

Keywords. Accelerated gradient algorithms; Nesterov accelerated gradient algorithm ; Ravine algorithm; Hessian driven damping; high-resolution ODE; convergence rates; Lyapunov analysis.

AMS Subject Classification. 37N40, 46N10, 49M30, 65B99, 65K05, 65K10, 90B50, 90C25.

1 Introduction

Given $\mathcal{H}$ a real Hilbert space, we consider the case of convex differentiable optimization

$$\min \{f(x) : x \in \mathcal{H}\}, \tag{1}$$

*Laboratoire XLIM, Université de Limoges, 123 Avenue Albert Thomas, 87060 Limoges CEDEX, France. Email: samir.adly@unilim.fr.

†IMAG, Université Montpellier, CNRS UMR 5149, Place Eugène Bataillon, 34095 Montpellier Cedex 5, France. E-mail: hedy.attouch@umontpellier.fr.

‡Normandie Univ-ENSICAEN, GREYC, CNRS UMR 6072, 14050 Caen Cedex France. Email: Jalal.Fadili@greyc.ensicaen.fr.

where $f : \mathcal{H} \to \mathbb{R}$ is a convex function of class $\mathcal{C}^1$, whose gradient ∇f is Lipschitz continuous, and which satisfies $\operatorname{argmin}_{\mathcal{H}}(f) \neq \emptyset$. In this setting, we revisit and compare three basic accelerated gradient methods from a dynamic perspective:

- The accelerated gradient method of Nesterov [30, 31, 32], (NAG) for short.
- The Ravine method of Gelfand and Tsetlin [25], (RAG) for short.
- The Inertial gradient algorithm with Hessian-driven damping, (IGAHD) for short, recently introduced by Attouch, Chbani, Fadili and Riahi [10].

This is the first time that a systematic comparative study of these algorithms is made. This has been made possible thanks to recent progress concerning the link between optimization algorithms and dissipative dynamical systems. By using the high-resolution ODE technique introduced by Shi, Du, Jordan, and Shi [36], we analyze the similarities and differences between these algorithms. This technique originates from numerical analysis, where high accuracy is required, particularly in the simulation of fluid dynamics and PDEs, where multiple scales and transitions occur in physical phenomena. In comparison to low-resolution ODEs, high-resolution ODEs are more accurate continuous-time surrogates for the corresponding algorithms, enabling a better characterization and understanding of the accelerated methods. For more details, we refer to [36, 37, 41]. While the method can be developed by considering many other algorithms, we limit ourselves to these three algorithms because of their importance, their relative simplicity, and to stay within a reasonable length of this article.

The importance of this topic lies in the fact that in data science, image processing, and statistical learning, the gradient descent method is one of the most popular numerical algorithms for minimizing a smooth function due to its simplicity. However, one of its drawbacks is its slow convergence rate. The acceleration of first-order methods is now an active research area in large-scale numerical optimization with many important and concrete real-world applications.

1.1 Dynamical systems associated with the three algorithms

Let us review the known results regarding the dynamical interpretation of the three algorithms.

a) In 1983, Nesterov introduced a momentum method, known in the literature as Nesterov accelerated gradient (NAG for short) [30]. To obtain a continuous ODE surrogate of the NAG algorithm, a crucial step was taken by Su-Boyd-Candès in 2016 [38]. They introduced an asymptotic vanishing damping coefficient of the form $\frac{\alpha}{t}$, where $\alpha > 0$ and $t > 0$ represents the time variable in the inertial system (AVD_α) below. In particular, for a general convex function f, the condition $\alpha > 3$ guarantees the asymptotic convergence rate of the values with a rate of $o\left(1/t^2\right)$, as well as the weak convergence of each trajectory toward an optimal solution [12]. Results in the algorithmic case can be found in [23]. The subcritical case $\alpha \leq 3$ has been considered in [5] and [13], which showed that the best convergence rates are achieved for $\alpha \geq 3$. In recent years, there has been an in-depth study linking the NAG method to inertial dynamics with vanishing viscous damping; see [7, 8, 9, 12, 21, 22, 38].

b) The above inertial systems may suffer from transverse oscillations, and it is desirable to dampen them. This is precisely the motivation behind the introduction of geometric Hessian-driven damping in [4]. Several recent studies have been devoted to inertial dynamics that combine asymptotic vanishing damping with geometric Hessian-driven damping (sometimes called Newton-type inertial dynamics). See, for example, [16, 17, 10, 36]. In turn, the corresponding algorithms,

among which (IGAHD) enjoys several favorable properties, introduce a correction term in the NAG method, which reduces the oscillatory aspects.
Note that the explicit form of the Hessian-driven damping was introduced in [10] and [36], while the implicit form was considered in [3]. Additionally, it should be noted that Hessian-driven damping can be combined with other types of damping, such as dry friction, as shown in [1, 2].

c) The Ravine method (RAG for short) was introduced by Gelfand and Tsetlin in 1961 [25]. It mimics the flow of water in mountains, first descending rapidly through small, steep ravines and then flowing along the main river in the valley. The method also models the transmission of nerve impulses. For a long time, the RAG and NAG methods were confused with each other, as both algorithms describe the evolution of different variables governed by similar equations. Only recently, connections between the two methods were brought to the forefront by Attouch and Fadili in [14]. It was shown that the RAG and NAG methods have the same dynamical interpretation and exhibit similar fast convergence properties. Indeed, the low-resolution ODE (in the sense of [36]) for both methods is given by the dynamics (AVD_α). However, their high-resolution ODE exhibits an additional Hessian-driven damping, which provides a more accurate dynamical interpretation of the two schemes. For a recent account of the Ravine method, see [34] and [39].

1.2 Main results

Our main goal and contribution in this paper is to gain a better understanding of the distinctions between the three algorithms and clarify that despite sharing some similarities, they are definitely different. Our contribution is both theoretical and numerical.

- In this paper, we highlight the similarities as well as the differences between NAG and RAG. Differentiating the two algorithms is important because, even if they are close, when they are used as basic blocks of splitting algorithms (such as proximal gradient method, primal dual methods, ADMM, etc.), they can give rise to clearly different algorithms. We refine the comparison and show that the two algorithms have a similar high-resolution ODE of order $h = \sqrt{s}$, but a different high-resolution ODE of order $h^2 = s$, where s is the step length in the gradient step. We also conduct some numerical experiments to support the theoretical part.

- We further refine the comparison between (IGAHD) and (NAG) (as well as with (RAG)) and show that although the two algorithms have a similar low-resolution ODE, their high-resolution ODE's are different. In other words, in the corresponding Taylor expansions with a time-step of h, the terms of order 0 are the same, but the terms of order 1 in h are different. This induces significant differences between the two algorithms, with a clear advantage of (IGAHD), which is confirmed by the numerical experiments.

- As a by-product of our analysis, we highlight new aspects concerning the subtle links between continuous dynamical systems and algorithms.

1.3 Organization of the paper

Each section in this paper is dedicated to one of the three algorithms: (NAG), (RAG), and (IGAHD). In each section, we review the algorithm's main convergence properties and provide its dynamical interpretation. We also perform a high-resolution analysis of the algorithm at order p. To avoid overly complicated computations, we set p to the minimum value that allows us to

distinguish the algorithm from the others. Section 2 covers (NAG), Section 3 covers (RAG), and Section 4 covers (IGAHD). In Section 5, we present some numerical experiments that illustrate the theoretical results using the performance profile. Finally, in Section 6, we discuss previous results.

2 NAG algorithm

Given α a positive parameter, the following second-order ODE

$$\ddot{x}(t) + \frac{\alpha}{t}\dot{x}(t) + \nabla f(x(t)) = 0, \tag{AVD$_\alpha$}$$

was introduced in [38]. An appropriate temporal discretized version of this ODE with step-size $s > 0$ gives the scheme (NAG)$_\alpha$, that we denote (NAG) for simplicity, and which reads

$$\begin{cases} y_k &= x_k + \left(1 - \frac{\alpha}{k}\right)(x_k - x_{k-1}) \\ x_{k+1} &= y_k - s\nabla f(y_k). \end{cases} \tag{NAG}$$

(NAG) performs a gradient step at y_k, which is an extrapolated point obtained from x_k and x_{k-1}. This is illustrated in Figure 1.

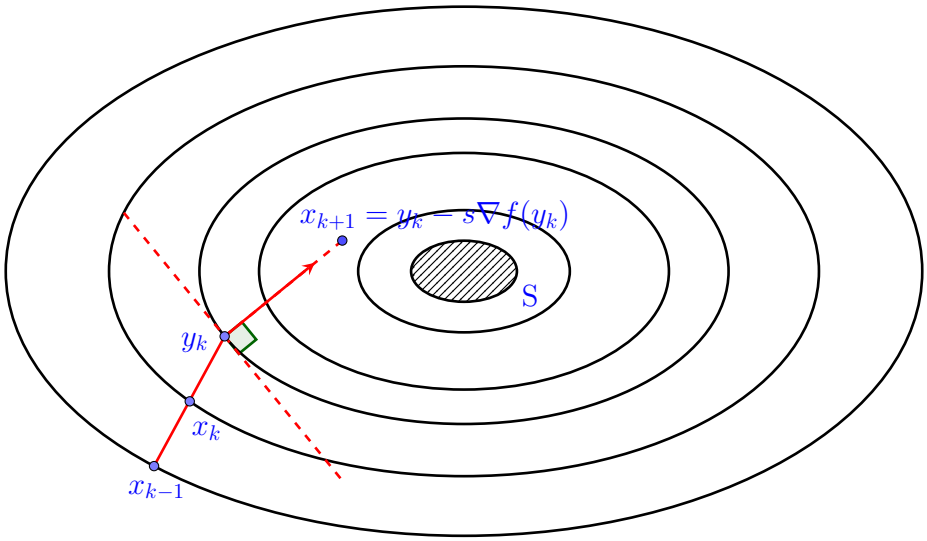

Figure 1: A geometrical illustration of (NAG).

In order to compare the (IGAHD) algorithm, which is related to but different from (NAG), it is important to specify the process of temporal discretization. Typically, the implicit scheme (proximal) preserves the convergence properties of the continuous dynamics from which it is derived. By using this scheme, we obtain

$$\frac{1}{s}(x_{k+1} - 2x_k + x_{k-1}) + \frac{\alpha}{ks}(x_k - x_{k-1}) + \nabla f(x_{k+1}) = 0,$$

which gives

$$x_{k+1} = \mathrm{prox}_{sf}(y_k),$$

where

$$y_k = x_k + \big(1 - \frac{\alpha}{k}\big)(x_k - x_{k-1}).$$

Then, replacing the proximal step by a gradient step gives the algorithm (NAG). Indeed, based on the definition of the proximal operator, for a differentiable convex function f, we can approximate $\mathrm{prox}_{sf}(x)$ for small values of s as $(\mathrm{Id} + s\nabla f)^{-1}(x) \simeq x - s\nabla f(x)$, for every $x \in \mathcal{H}$.

2.1 Convergence properties of NAG

The effectiveness of the (NAG) scheme depends crucially on the tuning of the extrapolation parameter α_k, which is given by $\alpha_k = 1 - \frac{\alpha}{k}$. This parameter tends to one from below in a subtle but controlled manner. The historical version of the scheme corresponds to $\alpha = 3$, which yields an asymptotic convergence rate of $f(x(t)) - \min_{\mathcal{H}} f = \mathcal{O}\left(\frac{1}{t^2}\right)$ for the continuous dynamics described by (AVD_α), and $f(x_k) - \min_{\mathcal{H}} f = \mathcal{O}\left(\frac{1}{k^2}\right)$ for the corresponding discretized scheme (NAG). We note that the Nesterov coefficient is asymptotically equivalent to $1 - 3/k$. By selecting $\alpha > 3$, every trajectory is shown to converge towards an optimal solution, and the convergence rate is improved to $f(x(t)) - \min_{\mathcal{H}} f = o\left(\frac{1}{t^2}\right)$ and $f(x_k) - \min_{\mathcal{H}} f = o\left(\frac{1}{k^2}\right)$, respectively.These results are established using Lyapunov analysis [12, 15, 23], which we summarize below.

Theorem 1. *Let $f : \mathcal{H} \to \mathbb{R}$ be a $\mathcal{C}^1$ convex function whose gradient ∇f is L-Lipschitz continuous, and $S := \mathrm{argmin}_{\mathcal{H}}(f) \neq \emptyset$. Let $x^\star \in S$. Take $\alpha \geq 3$, and $s \in]0, 1/L[$. Let $(x_k)_{k\in\mathbb{N}}$ be a sequence generated by the (NAG) algorithm. Set $t_k = \frac{k-1}{\alpha-1}$, and define, for each integer $k \geq 1$*

$$E_k := t_k^2(f(x_k) - \min_{\mathcal{H}} f) + \frac{1}{2s}\|x_{k-1} - x^\star + t_k(x_k - x_{k-1})\|^2.$$

Then, the sequence $(E_k)_{k\in\mathbb{N}}$ is nonincreasing, and as $k \to +\infty$

$$f(x_k) - \min_{\mathcal{H}} f = \mathcal{O}\left(\frac{1}{k^2}\right), \quad \|x_k - x_{k-1}\| = \mathcal{O}\left(\frac{1}{k}\right), \quad \sum_{k\in\mathbb{N}} k^2\|\nabla f(x_k)\|^2 < +\infty.$$

In addition, when $\alpha > 3$,

$$f(x_k) - \min f = o\left(\frac{1}{k^2}\right), \quad \|x_k - x_{k-1}\| = o\left(\frac{1}{k}\right), \quad \textit{and} \quad \text{w-lim}\, x_k = x^\star \in S,$$

where w-lim *stands for the weak limit.*

There is an extensive literature dedicated to studying these questions from various perspectives, providing an in-depth understanding of the (NAG) method. For instance, [6, 7, 8, 12, 13, 19, 23, 26, 27, 28, 29, 35, 36, 38, 40] have contributed significantly to this area of research. Interestingly, the fast convergence of the gradients satisfied by (NAG), as mentioned earlier, was recently discovered in [14].

2.2 Super-resolution ODE of NAG

The high-resolution method is extensively used in fluid mechanics, where physical phenomena occur at multiple scales; see e.g., [33]. In the high-resolution ODE of order p, the idea is not to let $h \to 0$, but to take into account the terms in h^p in the asymptotic expansions, and to discard the terms in h^{p+1} and higher. To make appear a difference between the high-resolution ODE of (NAG) and (RAG) we will take $p = 2$. In other words, we will take into account the terms in $s = h^2$ in the asymptotic expansions, and discard the terms in h^3 and higher. The resulting dynamic is called the super-resolution ODE. On the other hand to distinguish (NAG) and (RAG) from (IGAHD) the high resolution of order $p = 1$ is enough. This will reveal a clear difference between (NAG) and (RAG) from one hand, and (IGAHD) from the other hand.

Theorem 2. *Assume that f is $\mathcal{C}^3$. The super-resolution ODE with temporal step-size $\sqrt{s}$ of (NAG) gives the inertial dynamic with Hessian driven damping*

$$\ddot{X}(t) + \frac{\alpha}{t}\dot{X}(t) + \sqrt{s}\nabla^2 f(X(t))\dot{X}(t) + \left(1 + \frac{\alpha\sqrt{s}}{2t}\right)\nabla f(X(t)) \tag{2}$$
$$+\frac{s}{2}\Big(\frac{1}{6}X^{(4)}(t) + \frac{\alpha}{3t}\dddot{X}(t) - \frac{\alpha}{t}\nabla^2 f(X(t))\dot{X}(t) - \nabla^2 f(X(t))\ddot{X}(t) + \nabla^3 f(X(t))\big(\dot{X}(t), \dot{X}(t)\big)\Big) = 0.$$

When neglecting the terms of order higher or equal to 2 in (2), we recover the high-resolution ODE of (NAG) of order 1 which was obtained in [14], namely

$$\ddot{X}(t) + \frac{\alpha}{t}\dot{X}(t) + \sqrt{s}\nabla^2 f(X(t))\dot{X}(t) + \left(1 + \frac{\alpha\sqrt{s}}{2t}\right)\nabla f(X(t)) = 0. \tag{3}$$

Proof. First write (NAG) equivalently as

$$x_{k+1} = x_k + \left(1 - \frac{\alpha}{k}\right)(x_k - x_{k-1}) - s\nabla f\Big(x_k + \left(1 - \frac{\alpha}{k}\right)(x_k - x_{k-1})\Big).$$

Thus, with $s = h^2$, this is also equivalent to

$$\frac{x_{k+1} - 2x_k + x_{k-1}}{h^2} + \frac{\alpha}{kh}\frac{x_k - x_{k-1}}{h} + \nabla f(y_k) = 0. \tag{4}$$

For each $k \in \mathbb{N}$, let $t_k := h(k + c)$ for a real parameter c to be adjusted later. We use the ansatz that $x_k = X(t_k)$, where $t \mapsto X(t)$ is a smooth enough curve defined for $t \geq t_0 > 0$. Indeed, we know that such a smooth curve exists by, for example, taking a solution trajectory of the continuous dynamic (AVD_α). Our goal is to find a dynamic that best reflects the properties of the algorithm (NAG). Expanding the various quantities involved in (NAG) in powers of h, when h is close to zero, we obtain

$$X(t_k + h) = X(t_k) + h\dot{X}(t_k) + \frac{1}{2}h^2\ddot{X}(t_k) + \frac{1}{6}h^3\dddot{X}(t_k) + \frac{1}{24}h^4X^{(4)}(t_k) + \frac{1}{120}h^5X^{(5)}(t_k) + \mathcal{O}(h^6) \tag{5}$$
$$X(t_k - h) = X(t_k) - h\dot{X}(t_k) + \frac{1}{2}h^2\ddot{X}(t_k) - \frac{1}{6}h^3\dddot{X}(t_k) + \frac{1}{24}h^4X^{(4)}(t_k) - \frac{1}{120}h^5X^{(5)}(t_k) + \mathcal{O}(h^6) \tag{6}$$

where $X^{(p)}$ stands for the p-order time derivative of X. According to $x_{k+1} = X(t_k+h)$ and $x_{k-1} = X(t_k-h)$, by adding (5) and (6), we obtain

$$\frac{x_{k+1} - 2x_k + x_{k-1}}{h^2} = \ddot{X}(t_k) + \frac{1}{12}h^2X^{(4)}(t_k) + \mathcal{O}(h^4).$$

Moreover, (6) gives

$$\frac{x_k - x_{k-1}}{h} = \dot{X}(t_k) - \frac{1}{2}h\ddot{X}(t_k) + \frac{1}{6}h^2\dddot{X}(t_k) - \frac{1}{24}h^3 X^{(4)}(t_k) + \mathcal{O}(h^4).$$

We also have

$$\begin{aligned}
\nabla f(y_k) &= \nabla f\left(x_k + h\left(1-\frac{\alpha}{k}\right)\frac{x_k - x_{k-1}}{h}\right) \\
&= \nabla f\left(X(t_k) + h\left(1-\frac{\alpha}{k}\right)\left(\dot{X}(t_k) - \frac{1}{2}h\ddot{X}(t_k) + \mathcal{O}(h^2)\right)\right) \\
&= \nabla f\left(X(t_k) + h\left(1-\frac{\alpha}{k}\right)\left(\dot{X}(t_k) - \frac{1}{2}h\ddot{X}(t_k)\right) + \mathcal{O}(h^3)\right) \\
&= \nabla f(X(t_k)) + h\left(1-\frac{\alpha}{k}\right)\nabla^2 f(X(t_k))\left(\dot{X}(t_k) - \frac{1}{2}h\ddot{X}(t_k)\right) \\
&\quad + \frac{1}{2}h^2\left(1-\frac{\alpha}{k}\right)^2\nabla^3 f(X(t_k))\left(\dot{X}(t_k) - \frac{1}{2}h\ddot{X}(t_k), \dot{X}(t_k) - \frac{1}{2}h\ddot{X}(t_k)\right) + \mathcal{O}(h^3) \\
&= \nabla f(X(t_k)) + h\left(1-\frac{\alpha}{k}\right)\nabla^2 f(X(t_k))\left(\dot{X}(t_k) - \frac{1}{2}h\ddot{X}(t_k)\right) \\
&\quad + \frac{1}{2}h^2\left(1-\frac{\alpha}{k}\right)^2\nabla^3 f(X(t_k))\left(\dot{X}(t_k), \dot{X}(t_k)\right) + \mathcal{O}(h^3).
\end{aligned}$$

Plugging the above results into (4), we obtain

$$\begin{aligned}
&\ddot{X}(t_k) + \frac{1}{12}h^2 X^{(4)}(t_k) + \frac{\alpha}{kh}\left(\dot{X}(t_k) - \frac{1}{2}h\ddot{X}(t_k) + \frac{1}{6}h^2\dddot{X}(t_k) - \frac{1}{24}h^3 X^{(4)}(t_k)\right) \\
&+\nabla f(X(t_k)) + h\left(1-\frac{\alpha}{k}\right)\nabla^2 f(X(t_k))\left(\dot{X}(t_k) - \frac{1}{2}h\ddot{X}(t_k)\right) \\
&+\frac{1}{2}h^2\left(1-\frac{\alpha}{k}\right)^2\nabla^3 f(X(t_k))\left(\dot{X}(t_k), \dot{X}(t_k)\right) + \mathcal{O}(h^2) = 0.
\end{aligned}$$

Equivalently,

$$\begin{aligned}
&\frac{h^2}{12}\left(1-\frac{\alpha}{2k}\right)X^{(4)}(t_k) + \frac{\alpha h}{6k}\dddot{X}(t_k) + \left(1-\frac{\alpha}{2k}\right)\ddot{X}(t_k) + \frac{\alpha}{kh}\dot{X}(t_k) + \nabla f(X(t_k)) \\
&+ h\left(1-\frac{\alpha}{k}\right)\nabla^2 f(X(t_k))\left(\dot{X}(t_k) - \frac{1}{2}h\ddot{X}(t_k)\right) + \frac{1}{2}h^2\left(1-\frac{\alpha}{k}\right)^2\nabla^3 f(X(t_k))\left(\dot{X}(t_k), \dot{X}(t_k)\right) + \mathcal{O}(h^3) = 0.
\end{aligned}$$

Dividing by $\left(1-\frac{\alpha}{2k}\right)$ gives

$$\begin{aligned}
&\frac{h^2}{12}X^{(4)}(t_k) + \frac{\alpha h^2}{6h\left(k-\frac{\alpha}{2}\right)}\dddot{X}(t_k) + \ddot{X}(t_k) + \frac{\alpha}{h\left(k-\frac{\alpha}{2}\right)}\dot{X}(t_k) + \left(1 + \frac{\alpha h}{2h\left(k-\frac{\alpha}{2}\right)}\right)\nabla f(X(t_k)) \\
&+h\left(1 - \frac{\frac{\alpha h}{2}}{h\left(k-\frac{\alpha}{2}\right)}\right)\nabla^2 f(X(t_k))\left(\dot{X}(t_k) - \frac{1}{2}h\ddot{X}(t_k)\right) \\
&+\frac{1}{2}h^2\left(1-\frac{\alpha}{k}\right)\left(1 - \frac{\frac{\alpha h}{2}}{h\left(k-\frac{\alpha}{2}\right)}\right)\nabla^3 f(X(t_k))\left(\dot{X}(t_k), \dot{X}(t_k)\right) + \mathcal{O}(h^3) = 0.
\end{aligned}$$

Set $c = -\frac{\alpha}{2}$ and thus $t_k := h\left(k - \frac{\alpha}{2}\right)$. We obtain

$$\begin{aligned}
&\frac{h^2}{12}X^{(4)}(t_k) + \frac{\alpha h^2}{6t_k}\dddot{X}(t_k) + \ddot{X}(t_k) + \frac{\alpha}{t_k}\dot{X}(t_k) + \left(1 + \frac{\alpha h}{2t_k}\right)\nabla f(X(t_k)) \\
&+h\left(1 - \frac{\alpha h}{2t_k}\right)\nabla^2 f(X(t_k))\left(\dot{X}(t_k) - \frac{1}{2}h\ddot{X}(t_k)\right) \\
&+\frac{1}{2}h^2\left(1 - \frac{\alpha h}{t_k + \frac{\alpha h}{2}}\right)\left(1 - \frac{\alpha h}{2t_k}\right)\nabla^3 f(X(t_k))\left(\dot{X}(t_k), \dot{X}(t_k)\right) + \mathcal{O}(h^3) = 0.
\end{aligned}$$

Keeping only the terms of order less than or equal to 2 we obtain

$$\begin{aligned}
&\frac{h^2}{12}X^{(4)}(t_k) + \frac{\alpha h^2}{6t_k}\dddot{X}(t_k) + \ddot{X}(t_k) + \frac{\alpha}{t_k}\dot{X}(t_k) + \left(1 + \frac{\alpha h}{2t_k}\right)\nabla f(X(t_k)) \\
&+h\left(1 - \frac{\alpha h}{2t_k}\right)\nabla^2 f(X(t_k))\dot{X}(t_k) - \frac{h^2}{2}\nabla^2 f(X(t_k))\ddot{X}(t_k) \\
&+\frac{h^2}{2}\nabla^3 f(X(t_k))\left(\dot{X}(t_k), \dot{X}(t_k)\right) + \mathcal{O}(h^3) = 0.
\end{aligned}$$

Equivalently

$$\begin{aligned}
&\ddot{X}(t_k) + \frac{\alpha}{t_k}\dot{X}(t_k) + \left(1 + \frac{\alpha\sqrt{s}}{2t_k}\right)\nabla f(X(t_k)) + \sqrt{s}\nabla^2 f(X(t_k))\dot{X}(t) \\
&+\frac{s}{2}\Big(\frac{1}{6}X^{(4)}(t_k) + \frac{\alpha}{3t_k}\dddot{X}(t_k) - \frac{\alpha}{t_k}\nabla^2 f(X(t_k))\dot{X}(t_k) - \nabla^2 f(X(t_k))\ddot{X}(t_k) + \nabla^3 f(X(t_k))\left(\dot{X}(t_k), \dot{X}(t_k)\right)\Big) \\
&+\mathcal{O}(h^3) = 0.
\end{aligned}$$

We thus obtain the claimed inertial dynamic with Hessian driven damping. □

3 The Ravine algorithm

The Ravine algorithm, RAG for short, generates sequences $(y_k)_{k\in\mathbb{N}}$ which satisfy

$$\begin{cases} w_k = y_k - s\nabla f(y_k) \\ y_{k+1} = w_k + \left(1 - \frac{\alpha}{k+1}\right)(w_k - w_{k-1}). \end{cases} \tag{RAG}$$

Figure 2 depicts a simplified representation of water flow in the mountains, where the water first descends rapidly through small steep ravines and then flows along the main river in the valley. This representation was the primary motivation for the algorithm and its terminology. As the figure suggests, to better match the physical interpretation, it would be interesting to consider ravines with several gradient steps. Similarly, it could be interesting to extrapolate with $p \geq 2$ ravines instead of just two.

Historically, the Ravine method was introduced with a fixed extrapolation coefficient. Taking the extrapolation coefficient equal to $\left(1 - \frac{\alpha}{k+1}\right)$ makes (RAG) in accordance with (NAG) and is crucial to obtain an accelerated method. A geometric interpretation of (RAG) on a quadratic function is given in Fig. 3.

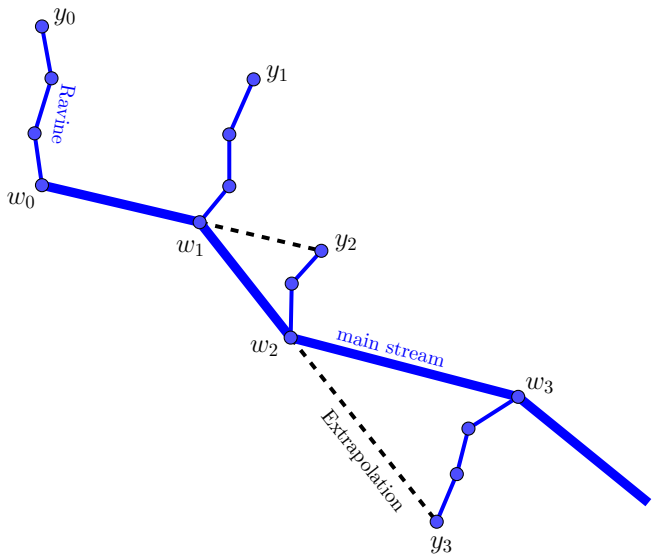

Figure 2: Interpretation of the Ravine method

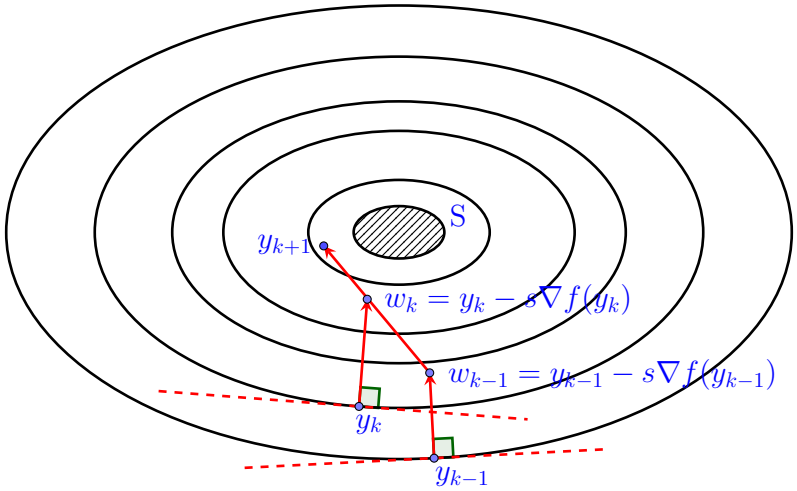

Figure 3: A geometrical illustration of (RAG).

The Lyapunov analysis of (RAG) is based on the energy sequences $(E_k)_{k\in\mathbb{N}}$: for $x^\star \in \operatorname{argmin}_{\mathcal{H}}(f)$,

$$E_k := h^2(k+2-\alpha)(k+1)(f(y_k)-f(x^\star)) + \frac{1}{2}\|z_k\|^2 \tag{7}$$

$$z_k := (\alpha-1)(y_{k+1}-x^\star) + h(k+2-\alpha)\Big(v_k + h\nabla f(y_k)\Big). \tag{8}$$

Theorem 3 ([14]). *Let $f : \mathcal{H} \to \mathbb{R}$ be a $\mathcal{C}^1$ convex function whose gradient is L-Lipschitz continuous, and $S := \operatorname{argmin}_{\mathcal{H}}(f) \neq \emptyset$. Let $(y_k)_{k\in\mathbb{N}}$ be the sequence generated by (RAG), where $\alpha \geq 3$ and $sL < 1$. Then the sequence $(E_k)_{k\in\mathbb{N}}$ defined by (7)–(8) is nonincreasing for $k \geq 2\alpha - 3$, and the*

following convergence rates are satisfied:

(i) $f(y_k) - \min_{\mathcal{H}} f = \mathcal{O}\left(\frac{1}{k^2}\right), \quad \|y_k - y_{k-1}\| = \mathcal{O}\left(\frac{1}{k}\right).$

(ii) $\sum_{k\in\mathbb{N}} k^2 \|\nabla f(y_k)\|^2 < +\infty.$

In addition, when $\alpha > 3$,

(iii) $f(y_k) - \min f = o\left(\frac{1}{k^2}\right)$, $\|y_k - y_{k-1}\| = o\left(\frac{1}{k}\right)$ *and* w-lim $y_k = y^\star \in S$ *where* w-lim *stands for the weak limit.*

(iv) $\sum_{k\in\mathbb{N}} k(f(y_k) - f(x^\star)) < +\infty.$

3.1 First comparison results between (NAG) and (RAG)

At the origin of the confusion between the two methods is the fact that they can be defined by the same equations. But, they describe the evolution of different variables. Specifically, the variable y_k which enters the definition of (NAG) follows the (RAG) algorithm.
We notice that if $(x_k)_{k\in\mathbb{N}}$ is a sequence generated by the algorithm (NAG), by defining the associated sequence $(y_k)_{k\in\mathbb{N}}$ by

$$y_k = x_k + \left(1 - \frac{\alpha}{k}\right)(x_k - x_{k-1}).$$

Then, it is easy to show that $(y_k)_{k\in\mathbb{N}}$ follows the algorithm (RAG).
Conversely, if $(y_k)_{k\in\mathbb{N}}$ is a sequence generated by (RAG), then the sequence $(x_k)_{k\in\mathbb{N}}$ defined by

$$x_{k+1} = y_k - s\nabla f(y_k)$$

satisfies (NAG). For more details, we refer to [14, Theorem 2.2].

3.2 Super resolution ODE of RAG

Our approach is similar to the one developed in the previous section, as shown precisely in the following theorem.

Theorem 4. *Assume that* f *is* $\mathcal{C}^3$. *The super-resolution ODE with temporal step-size* $\sqrt{s}$ *of* (RAG) *gives the inertial dynamic with Hessian driven damping*

$$\ddot{Y}(t) + \frac{\alpha}{t}\dot{Y}(t) + \sqrt{s}\nabla^2 f(Y(t))\dot{Y}(t) + \left(1 + \frac{\alpha\sqrt{s}}{2t}\right)\nabla f(Y(t)) \tag{9}$$
$$+\frac{s}{2}\left(\frac{1}{6}Y^{(4)}(t) + \frac{\alpha}{3t}\dddot{Y}(t) - \frac{\alpha}{t}\nabla^2 f(Y(t))\dot{Y}(t) - \nabla^2 f(Y(t))\ddot{Y}(t) - \nabla^3 f(Y(t))\big(\dot{Y}(t), \dot{Y}(t)\big)\right) = 0.$$

When neglecting the terms of order higher or equal to 2 *in* (9), *we recover the high-resolution ODE of* (RAG) *of order* 1 *which was obtained in [14], namely*

$$\ddot{Y}(t) + \frac{\alpha}{t}\dot{Y}(t) + \sqrt{s}\nabla^2 f(Y(t))\dot{Y}(t) + \left(1 + \frac{\alpha\sqrt{s}}{2t}\right)\nabla f(Y(t)) = 0. \tag{10}$$

Proof. According to (RAG), we have

$$\begin{aligned} y_{k+1} &= y_k - s\nabla f(y_k) + \left(1 - \frac{\alpha}{k+1}\right)\Big(y_k - s\nabla f(y_k) - (y_{k-1} - s\nabla f(y_{k-1}))\Big) \\ &= y_k + \left(1 - \frac{\alpha}{k+1}\right)(y_k - y_{k-1}) - s\nabla f(y_k) - s\left(1 - \frac{\alpha}{k+1}\right)\Big(\nabla f(y_k) - \nabla f(y_{k-1})\Big). \end{aligned}$$

Dividing by $s = h^2$, we equivalently obtain,

$$\frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} + \frac{\alpha}{kh+h}\frac{y_k - y_{k-1}}{h} + \nabla f(y_k) + \left(1 - \frac{\alpha}{k+1}\right)(\nabla f(y_k) - \nabla f(y_{k-1})) = 0. \quad (11)$$

Let us arrange the above formula, so as to prepare it for its analysis by Taylor expansion. After multiplying (11) by $\frac{k+1}{k+1-\alpha}$, we get

$$\begin{aligned} \frac{k+1}{k+1-\alpha}\frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} + \frac{\alpha}{k+1-\alpha}\frac{y_k - y_{k-1}}{h^2} + \frac{k+1}{k+1-\alpha}\nabla f(y_k) \\ + \nabla f(y_k) - \nabla f(y_{k-1}) = 0. \quad (12) \end{aligned}$$

Notice then that

$$\frac{y_k - y_{k-1}}{h^2} = \frac{y_{k+1} - y_k}{h^2} - \frac{y_{k+1} - 2y_k + y_{k-1}}{h^2}.$$

Thus, (12) can be formulated equivalently as follows

$$\begin{aligned} \Big(\frac{k+1}{k+1-\alpha} - \frac{\alpha}{k+1-\alpha}\Big)\frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} + \frac{\alpha}{k+1-\alpha}\frac{y_{k+1} - y_k}{h^2} \\ + \frac{k+1}{k+1-\alpha}\nabla f(y_k) + \nabla f(y_k) - \nabla f(y_{k-1}) = 0. \end{aligned}$$

After reduction we arrive at

$$\begin{aligned} \frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} + \frac{\alpha}{(k+1-\alpha)h}\frac{y_{k+1} - y_k}{h} + \left(1 + \frac{\alpha}{k+1-\alpha}\right)\nabla f(y_k) \\ + \nabla f(y_k) - \nabla f(y_{k-1}) = 0. \quad (13) \end{aligned}$$

Building on (13), we now follow a device similar to the one developed in the previous section. For each $k \in \mathbb{N}$, set $t_k := (k+c)h$, where c is a real parameter that will be adjusted later. We use the ansatz that $y_k = Y(t_k)$ for some smooth enough curve $t \mapsto X(t)$ defined for $t \geq t_0 > 0$. Performing a Taylor expansion in powers of h, when h is close to zero, of the different quantities involved in (13), we obtain

$$Y(t_{k+1}) = Y(t_k) + h\dot{Y}(t_k) + \frac{1}{2}h^2\ddot{Y}(t_k) + \frac{1}{6}h^3\dddot{Y}(t_k) + \frac{1}{24}h^4Y^{(4)}(t_k) + \frac{1}{120}h^5Y^{(5)}(t_k) + \mathcal{O}(h^6) \quad (14)$$

$$Y(t_{k-1}) = Y(t_k) - h\dot{Y}(t_k) + \frac{1}{2}h^2\ddot{Y}(t_k) - \frac{1}{6}h^3\dddot{Y}(t_k) + + \frac{1}{24}h^4Y^{(4)}(t_k) - \frac{1}{120}h^5Y^{(5)}(t_k) + \mathcal{O}(h^6) \quad (15)$$

where $Y^{(p)}$ stands for the p-order time derivative of Y. According to $y_{k+1} = Y(t_k + h)$ and $y_{k-1} = Y(t_k - h)$, by adding (14) and (15) we obtain

$$\frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} = \ddot{Y}(t_k) + \frac{1}{12}h^2Y^{(4)}(t_k) + \mathcal{O}(h^4).$$

Moreover, (14) gives

$$\frac{y_{k+1}-y_k}{h} = \dot{Y}(t_k) + \frac{1}{2}h\ddot{Y}(t_k) + \frac{1}{6}h^2\dddot{Y}(t_k) + \mathcal{O}(h^4).$$

By Taylor expansion of $\nabla f \circ Y$ we have

$$\begin{aligned}
\nabla f(y_k) - \nabla f(y_{k-1}) &= (\nabla f \circ Y)(t_k) - (\nabla f \circ Y)(t_{k-1}) \\
&= \frac{d}{dt}(\nabla f \circ Y)(t_k))h - \frac{1}{2}\frac{d^2}{dt^2}(\nabla f \circ Y)(t_k))h^2 + \mathcal{O}\left(h^3\right) \qquad (16)\\
&= \frac{d}{dt}(\nabla f \circ Y)(t_k))h - \frac{1}{2}\nabla^3 f(Y(t_k))\big(\dot{Y}(t_k), \dot{Y}(t_k)\big)h^2 \\
&\quad -\frac{1}{2}\nabla^2 f(Y(t_k))\ddot{Y}(t_k)h^2 + \mathcal{O}\left(h^3\right),
\end{aligned}$$

thus giving rise to $\nabla^3 f$, a tensor of order 3 which will play an important role in distinguishing the two algorithms from each other. Indeed we will continue the calculation with the more compact formulation (16). Plugging all of the above results into (13), we obtain

$$\begin{aligned}
&\big(\ddot{Y}(t_k) + \frac{1}{12}h^2 Y^{(4)}(t_k) + \mathcal{O}(h^3)\big) + \frac{\alpha}{(k+1-\alpha)h}\big(\dot{Y}(t_k) + \frac{1}{2}h\ddot{Y}(t_k) + \frac{1}{6}h^2\dddot{Y}(t_k) + \mathcal{O}(h^3)\big) \\
&\quad + \frac{k+1}{k+1-\alpha}\nabla f(Y(t_k)) + \frac{d}{dt}(\nabla f \circ Y)(t_k))h - \frac{1}{2}\frac{d^2}{dt^2}(\nabla f \circ Y)(t_k))h^2 + \mathcal{O}\left(h^3\right) = 0.
\end{aligned}$$

After multiplication by $\frac{(k+1-\alpha)h}{\alpha}$, and reduction of the terms, we obtain

$$\begin{aligned}
&\frac{h}{\alpha}\left(k+1-\frac{\alpha}{2}\right)\ddot{Y}(t_k) + \dot{Y}(t_k) + \frac{(k+1)h}{\alpha}\nabla f(Y(t_k)) + h\frac{(k+1-\alpha)h}{\alpha}\nabla^2 f(Y(t_k))\dot{Y}(t_k) \\
&+\frac{(k+1-\alpha)h}{\alpha}\left(\frac{1}{12}h^2 Y^{(4)}(t_k) - \frac{1}{2}\frac{d^2}{dt^2}(\nabla f \circ Y)(t_k))h^2\right) + \frac{1}{6}h^2\dddot{Y}(t_k) + \mathcal{O}(h^3) = 0.
\end{aligned}$$

Dividing by $\frac{h}{\alpha}(k+1-\frac{\alpha}{2})$ yields

$$\begin{aligned}
&\ddot{Y}(t_k) + \frac{\alpha}{(k+1-\frac{\alpha}{2})h}\dot{Y}(t_k) + \left(1 + \frac{\frac{\alpha}{2}}{k+1-\frac{\alpha}{2}}\right)\nabla f(Y(t_k)) + h\left(1 - \frac{\frac{\alpha}{2}}{k+1-\frac{\alpha}{2}}\right)\nabla^2 f(Y(t_k))\dot{Y}(t_k) \\
&+\left(1 - \frac{\frac{\alpha}{2}}{k+1-\frac{\alpha}{2}}\right)\left(\frac{1}{12}h^2 Y^{(4)}(t_k) - \frac{1}{2}\frac{d^2}{dt^2}(\nabla f \circ Y)(t_k))h^2\right) + \frac{\alpha}{(k+1-\frac{\alpha}{2})h}\frac{1}{6}h^2\dddot{Y}(t_k) + \mathcal{O}(h^3) = 0.
\end{aligned}$$

Take $c = 1 - \frac{\alpha}{2}$ and thus $t_k := (k+1-\frac{\alpha}{2})h$. We obtain

$$\begin{aligned}
&\ddot{Y}(t_k) + \frac{\alpha}{t_k}\dot{Y}(t_k) + \left(1 + \frac{\alpha h}{2t_k}\right)\nabla f(Y(t_k)) + h\left(1 - \frac{\alpha h}{2t_k}\right)\nabla^2 f(Y(t_k))\dot{Y}(t_k) \\
&\quad + \left(1 - \frac{\alpha h}{2t_k}\right)\left(\frac{1}{12}h^2 Y^{(4)}(t_k) - \frac{1}{2}\frac{d^2}{dt^2}(\nabla f \circ Y)(t_k))h^2\right) + \frac{\alpha h^2}{6t_k}\dddot{Y}(t_k) + \mathcal{O}(h^3) = 0.
\end{aligned}$$

After reduction

$$\begin{aligned}
&\ddot{Y}(t_k) + \frac{\alpha}{t_k}\dot{Y}(t_k) + \left(1 + \frac{\alpha h}{2t_k}\right)\nabla f(Y(t_k)) + h\left(1 - \frac{\alpha h}{2t_k}\right)\nabla^2 f(Y(t_k))\dot{Y}(t_k) \\
&\quad + \frac{1}{12}h^2 Y^{(4)}(t_k) - \frac{1}{2}\frac{d^2}{dt^2}(\nabla f \circ Y)(t_k))h^2 + \frac{\alpha h^2}{6t_k}\dddot{Y}(t_k) + \mathcal{O}(h^3) = 0.
\end{aligned}$$

Equivalently

$$\ddot{Y}(t_k) + \frac{\alpha}{t_k}\dot{Y}(t_k) + \left(1 + \frac{\alpha h}{2t_k}\right)\nabla f(Y(t_k)) + h\nabla^2 f(Y(t_k))\dot{Y}(t_k)$$
$$+ \frac{1}{2}h^2\left(\frac{1}{6}Y^{(4)}(t_k) + \frac{\alpha}{3t_k}\dddot{Y}(t_k) - \frac{\alpha}{t_k}\nabla^2 f(Y(t_k))\dot{Y}(t_k) - \nabla^3 f(Y(t_k))\big(\dot{Y}(t_k), \dot{Y}(t_k)\big) - \nabla^2 f(Y(t_k))\ddot{Y}(t_k)\right)$$
$$+ \mathcal{O}(h^3) = 0.$$

By neglecting the term in h^3, and keeping the terms of order less than or equal to 2, we obtain the claimed inertial dynamic with Hessian driven damping. This completes the proof. □

The Ravine method, a precursor to accelerated gradient methods, has historically received limited attention. However, it has recently gained prominence in current research, notably through the presentation of Polyak [34]. It is worth noting that the Ravine method has occasionally been mistakenly associated with Nesterov Accelerated Gradient (NAG) method in the literature.
In light of our analysis presented in the current and previous sections, using a super-resolution ODE framework, we emphasize the distinctive characteristics of these two algorithms. The following theorem serves to highlight the disparities between them and is a direct consequence of Theorem 2 and Theorem 4. To complement this theoretical approach, we also refer to the numerical experiments section, where we compared the performance profiles of these algorithms. We refer also to Example 4, that shows how the distinct super-resolution ODEs associated with each algorithm generate different trajectories.

Theorem 5. *Nesterov Accelerated Gradient method* (NAG) *and Ravine Accelerated Gradient method* (RAG) *are fundamentally different optimization algorithms, characterized by their distinct super-resolution ODE* (2) *and* (9) *respectively.*

4 IGAHD algorithm

Consider a convex function f with a L-Lipschitz continuous gradient. The (IGAHD) algorithm arises from a temporal discretization of the dynamic system

$$\ddot{x}(t) + \frac{\alpha}{t}\dot{x}(t) + \beta\nabla^2 f(x(t))\dot{x}(t) + \left(1 + \frac{\beta}{t}\right)\nabla f(x(t)) = 0, \qquad \text{(DIN-AVD}_{\alpha,\beta,1+\frac{\beta}{t}}\text{)}$$

with damping parameters $\alpha \geq 3$ and $\beta \geq 0$. As Theorem 2 shows, this dynamic naturally arises as the high-resolution ODE of (NAG). We will demonstrate that a temporal discretization of (DIN-AVD$_{\alpha,\beta,1+\frac{\beta}{t}}$), similar to the one previously developed for (AVD$_\alpha$), gives the (IGAHD)$_{\alpha,\beta}$ algorithm, , which we will refer to as (IGAHD) for simplicity. This explains why (IGAHD) outperforms (NAG). The dynamic (DIN-AVD$_{\alpha,\beta,1+\frac{\beta}{t}}$) has better gradient convergence properties than (AVD$_\alpha$). This comparison is justified since both algorithms are obtained through a similar procedure. We will further discuss this point later. This exemplifies the interplay between continuous and discrete dynamical systems (algorithms) and the resulting mutual enrichment. The dynamic (DIN-AVD$_{\alpha,\beta,1+\frac{\beta}{t}}$) is closely related to the inertial system

$$\ddot{x}(t) + \frac{\alpha}{t}\dot{x}(t) + \beta\nabla^2 f(x(t))\dot{x}(t) + \nabla f(x(t)) = 0, \qquad \text{(DIN-AVD}_{\alpha,\beta}\text{)}$$

which was introduced in [16]. It combines the viscous damping with vanishing coefficient $\frac{\alpha}{t}$ with the geometric damping driven by the Hessian. Its formulation looks at a first glance more complicated than (AVD$_\alpha$). In [17], Attouch-Peypouquet-Redont showed that (DIN-AVD$_{\alpha,\beta}$) is equivalent to the first-order system in time and space

$$\begin{cases} \dot{x}(t) + \beta\nabla f(x(t)) - \left(\frac{1}{\beta} - \frac{\alpha}{t}\right) x(t) + \frac{1}{\beta} y(t) = 0; \\ \dot{y}(t) - \left(\frac{1}{\beta} - \frac{\alpha}{t} + \frac{\alpha\beta}{t^2}\right) x(t) + \frac{1}{\beta} y(t) = 0. \end{cases}$$

This provides a natural extension to $f : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$ proper lower semicontinuous and convex, just replacing the gradient ∇f by the convex subdifferential ∂f. Consider the time discretization of (DIN-AVD$_{\alpha,\beta,1+\frac{\beta}{t}}$)

$$\frac{1}{s}(x_{k+1} - 2x_k + x_{k-1}) + \frac{\alpha}{ks}(x_k - x_{k-1}) + \frac{\beta}{\sqrt{s}}(\nabla f(x_k) - \nabla f(x_{k-1}))$$
$$+ \frac{\beta}{k\sqrt{s}}\nabla f(x_{k-1}) + \nabla f(y_k) = 0,$$

with y_k inspired by Nesterov's accelerated scheme. We obtain the following scheme:

(IGAHD) : Inertial Gradient Algorithm with Hessian Damping.

Step k: $\alpha_k = 1 - \frac{\alpha}{k}$.

$$\begin{cases} y_k = x_k + \alpha_k(x_k - x_{k-1}) - \beta\sqrt{s}\,(\nabla f(x_k) - \nabla f(x_{k-1})) - \frac{\beta\sqrt{s}}{k}\nabla f(x_{k-1}) \\ x_{k+1} = y_k - s\nabla f(y_k). \end{cases}$$

This algorithm can be derived through a discretization procedure similar to that of (NAG). Specifically, it involves considering the implicit (proximal) temporal discretization of (DIN-AVD$_{\alpha,\beta,1+\frac{\beta}{t}}$)

$$\frac{1}{s}(x_{k+1} - 2x_k + x_{k-1}) + \frac{\alpha}{ks}(x_k - x_{k-1}) + \frac{\beta}{\sqrt{s}}(\nabla f(x_k) - \nabla f(x_{k-1}))$$
$$+ \frac{\beta}{k\sqrt{s}}\nabla f(x_{k-1}) + \nabla f(x_{k+1}) = 0.$$

Solving this equation with respect to x_{k+1} gives

$$x_{k+1} = \mathrm{prox}_{sf}(y_k).$$

Then, replacing the proximal step by a gradient step gives (IGAHD).

4.1 Convergence properties of (IGAHD)

The following results have been obtained in [10]. Following [8], set $t_{k+1} = \frac{k}{\alpha-1}$, whence $t_k = 1 + t_{k+1}\alpha_k$.

Given $x^\star \in \operatorname{argmin} f$, the Lyapunov analysis of (IGAHD) is based on the sequence $(E_k)_{k\in\mathbb{N}}$

$$E_k := t_k^2(f(x_k) - f(x^\star)) + \frac{1}{2s}\|v_k\|^2 \tag{17}$$

$$v_k := (x_{k-1} - x^\star) + t_k\Big(x_k - x_{k-1} + \beta\sqrt{s}\nabla f(x_{k-1})\Big). \tag{18}$$

Theorem 6. *Let $f : \mathcal{H} \to \mathbb{R}$ be a convex function whose gradient is L-Lipschitz continuous. Let $(x_k)_{k\in\mathbb{N}}$ be a sequence generated by algorithm (IGAHD), where $\alpha \geq 3$, $0 \leq \beta < 2\sqrt{s}$ and $sL \leq 1$. Then the sequence $(E_k)_{k\in\mathbb{N}}$ defined by (17)-(18) is non-increasing, and the following convergence rates are satisfied:*

(i) $f(x_k) - \min_{\mathcal{H}} f = \mathcal{O}\left(\dfrac{1}{k^2}\right)$ *as* $k \to +\infty$;

(ii) *Suppose that $\beta > 0$. Then*

$$\sum_k k^2\|\nabla f(y_k)\|^2 < +\infty \text{ and } \sum_k k^2\|\nabla f(x_k)\|^2 < +\infty.$$

The proof relies on Lyapunov analysis, using the sequence (E_k) and the following reinforced version of the gradient descent lemma which is valid for convex differentiable functions, see the Appendix of [10]. When $s \leq \frac{1}{L}$, and ∇f is L-lipschitz continuous, then for all $x, y \in \mathcal{H}$

$$f(y - s\nabla f(y)) \leq f(x) + \langle \nabla f(y), y - x\rangle - \frac{s}{2}\|\nabla f(y)\|^2 - \frac{s}{2}\|\nabla f(x) - \nabla f(y)\|^2.$$

The above convergence results are completed in [11] where it is shown the weak convergence of the iterates when $\alpha > 3$.

4.2 High resolution ODE of (IGAHD)

We adopt a similar approach as in Theorem 2 to derive a high resolution ODE for (IGAHD). The statement is presented in the following Theorem.

Theorem 7. *Assume that f is $\mathcal{C}^3$. The high-resolution ODE with temporal step-size $\sqrt{s}$ of the algorithm* (IGAHD) *gives the inertial dynamic with Hessian driven damping*

$$\ddot{X}(t) + \frac{\alpha}{t}\dot{X}(t) + \beta\nabla^2 f(X(t))\dot{X}(t) + \left(1 + \frac{\beta}{t}\right)\nabla f(X(t))$$
$$+\sqrt{s}\left(\frac{\alpha}{2t}\nabla f(X(t)) + \left(1 - \frac{\beta(\alpha-2)}{2t}\right)\nabla^2 f(X(t))\dot{X}(t) - \frac{\beta}{2}\nabla^3 f(X(t))\big(\dot{X}(t), \dot{X}(t)\big) - \frac{\beta}{2}\nabla^2 f(X(t))\ddot{X}(t)\right) = 0.$$

When $\beta = 0$ we recover the high-resolution ODE of (NAG) *of order* 1

$$\ddot{X}(t) + \frac{\alpha}{t}\dot{X}(t) + \left(1 + \frac{\alpha\sqrt{s}}{2t}\right)\nabla f(X(t)) + \sqrt{s}\nabla^2 f(X(t))\dot{X}(t) = 0.$$

Proof. First write (IGAHD) equivalently as

$$\frac{1}{s}(x_{k+1} - 2x_k + x_{k-1}) + \frac{\alpha}{ks}(x_k - x_{k-1}) + \frac{\beta}{\sqrt{s}}\left(1 - \frac{1}{k}\right)(\nabla f(x_k) - \nabla f(x_{k-1}))$$
$$+\frac{\beta}{k\sqrt{s}}\nabla f(x_k) + \nabla f(y_k) = 0. \tag{19}$$

For each $k \in \mathbb{N}$, set $t_k := h(k+c)$ for a real parameter c to be adjusted later, and use the ansatz that $x_k = X(t_k)$ for some smooth enough curve $t \mapsto X(t)$ defined for $t \geq t_0 > 0$. Indeed, we know that such a smooth curve exists, by taking for example a solution trajectory of the continuous dynamic (DIN-AVD$_{\alpha,\beta,1+\frac{\beta}{t}}$). What we are looking for is a dynamic that best reflects the properties of the algorithm (IGAHD).
Let us perform a Taylor expansion in powers of h, when h is close to zero, of the different quantities involved in (19). We first have

$$X(t_k+h) = X(t_k) + h\dot{X}(t_k) + \frac{1}{2}h^2\ddot{X}(t_k) + \frac{1}{6}h^3\dddot{X}(t_k) + \frac{1}{24}h^4X^{(4)}(t_k) + \frac{1}{120}h^5X^{(5)}(t_k) + \mathcal{O}(h^6) \quad (20)$$

$$X(t_k-h) = X(t_k) - h\dot{X}(t_k) + \frac{1}{2}h^2\ddot{X}(t_k) - \frac{1}{6}h^3\dddot{X}(t_k) + \frac{1}{24}h^4X^{(4)}(t_k) - \frac{1}{120}h^5X^{(5)}(t_k) + \mathcal{O}(h^6) \quad (21)$$

where $X^{(p)}$ stands for the p-order time derivative of X. According to $x_{k+1} = X(t_k+h)$ and $x_{k-1} = X(t_k-h)$, by adding (20) and (21), we obtain

$$\frac{x_{k+1} - 2x_k + x_{k-1}}{h^2} = \ddot{X}(t_k) + \frac{1}{12}h^2X^{(4)}(t_k) + \mathcal{O}(h^4).$$

Moreover, (21) gives

$$\frac{x_k - x_{k-1}}{h} = \dot{X}(t_k) - \frac{1}{2}h\ddot{X}(t_k) + \frac{1}{6}h^2\dddot{X}(t_k) - \frac{1}{24}h^3X^{(4)}(t_k) + \mathcal{O}(h^4).$$

By Taylor expansion of $\nabla f \circ X$, we have

$$\begin{aligned}
\nabla f(x_k) - \nabla f(x_{k-1}) &= (\nabla f \circ X)(t_k) - (\nabla f \circ X)(t_{k-1}) \\
&= h\frac{d}{dt}(\nabla f \circ X)(t_k)) - \frac{h^2}{2}\frac{d^2}{dt^2}(\nabla f \circ X)(t_k)) + \mathcal{O}\left(h^3\right) \\
&= h\nabla^2 f(X(t_k))\dot{X}(t_k) - \frac{h^2}{2}\nabla^3 f(X(t_k))\big(\dot{X}(t_k), \dot{X}(t_k)\big) \\
&\quad -\frac{h^2}{2}\nabla^2 f(X(t_k))\ddot{X}(t_k) + \mathcal{O}\left(h^3\right),
\end{aligned} \quad (22)$$

thus giving rise to $\nabla^3 f$, a tensor of order 3 which will play an important role in distinguishing the two algorithms from each other. According to the definition of y_k and the above formula we obtain (we retain only the terms of order less than or equal to 2)

$$\begin{aligned}
y_k &= x_k + h\left(1 - \frac{\alpha}{k}\right)\frac{x_k - x_{k-1}}{h} - \beta h\left(1 - \frac{1}{k}\right)(\nabla f(x_k) - \nabla f(x_{k-1})) - \frac{\beta h}{k}\nabla f(x_k) \\
&= X(t_k) + h\left(1 - \frac{\alpha}{k}\right)\left(\dot{X}(t_k) - \frac{1}{2}h\ddot{X}(t_k) + \mathcal{O}(h^2)\right) \\
&- \beta h\left(1 - \frac{1}{k}\right)\left(h\nabla^2 f(X(t_k))\dot{X}(t_k) + \mathcal{O}(h^2)\right) - \frac{\beta h}{k}\nabla f(X(t_k)) \\
&= X(t_k) + h\left(1 - \frac{\alpha}{k}\right)\dot{X}(t_k) + \mathcal{O}(h^2).
\end{aligned}$$

Note that the term $\frac{\beta h}{k}\nabla f(X(t_k)) = \frac{\beta h^2}{kh}\nabla f(X(t_k)) \sim \frac{\beta h^2}{t_k}\nabla f(X(t_k))$ is actually of order two.
We thus have

$$\begin{aligned}
\nabla f(y_k) &= \nabla f\left(X(t_k) + h\left(1 - \frac{\alpha}{k}\right)\dot{X}(t_k) + \mathcal{O}(h^2)\right) \\
&= \nabla f(X(t_k)) + h\left(1 - \frac{\alpha}{k}\right)\nabla^2 f(X(t_k))\dot{X}(t_k) + \mathcal{O}(h^2).
\end{aligned}$$

Plugging the above results into (19) (we use again $kh \sim t_k$), we obtain

$$\begin{aligned}
&\ddot{X}(t_k) + \frac{\alpha}{kh}\left(\dot{X}(t_k) - \frac{1}{2}h\ddot{X}(t_k)\right)\\
&+\beta\left(1-\frac{1}{k}\right)\left(\nabla^2 f(X(t_k))\dot{X}(t_k) - \frac{h}{2}\frac{d^2}{dt^2}(\nabla f\circ X)(t_k))\right) + \frac{\beta}{kh}\nabla f(X(t_k))\\
&+\nabla f(X(t_k)) + h\left(1-\frac{\alpha}{k}\right)\nabla^2 f(X(t_k))\dot{X}(t_k) + \mathcal{O}(h^2) = 0.
\end{aligned}$$

Equivalently,

$$\begin{aligned}
&\left(1-\frac{\alpha}{2k}\right)\ddot{X}(t_k) + \frac{\alpha}{kh}\dot{X}(t_k) + \left(1+\frac{\beta}{kh}\right)\nabla f(X(t_k))\\
&\quad + \left(h\left(1-\frac{\alpha}{k}\right) + \beta\left(1-\frac{1}{k}\right)\right)\nabla^2 f(X(t_k))\dot{X}(t_k) - \frac{1}{2}\beta h\left(1-\frac{1}{k}\right)\frac{d^2}{dt^2}(\nabla f\circ X)(t_k)) + \mathcal{O}(h^2) = 0.
\end{aligned}$$

Dividing by $\left(1-\frac{\alpha}{2k}\right)$ gives

$$\begin{aligned}
&\ddot{X}(t_k) + \frac{\alpha}{h\left(k-\frac{\alpha}{2}\right)}\dot{X}(t_k)\\
+\quad &\frac{1}{(1-\frac{\alpha}{2k})}\left(1+\frac{\beta}{kh}\right)\nabla f(X(t_k)) + \frac{1}{(1-\frac{\alpha}{2k})}\left(h\left(1-\frac{\alpha}{k}\right) + \beta\left(1-\frac{1}{k}\right)\right)\nabla^2 f(X(t_k))\dot{X}(t_k)\\
&-\frac{1}{(1-\frac{\alpha}{2k})}\frac{1}{2}\beta h\left(1-\frac{1}{k}\right)\frac{d^2}{dt^2}(\nabla f\circ X)(t_k)) + \mathcal{O}(h^2) = 0.
\end{aligned}$$

Set $c = -\frac{\alpha}{2}$ and thus $t_k := h\left(k-\frac{\alpha}{2}\right)$. After elementary computation we get

$$\begin{aligned}
&\ddot{X}(t_k) + \frac{\alpha}{t_k}\dot{X}(t_k) + \left(1+\frac{\alpha h}{2t_k}\right)\left(1+\frac{\beta}{t_k}(1-\frac{h\alpha}{2t_k})\right)\nabla f(X(t_k))\\
&+\left(1+\frac{\alpha h}{2t_k}\right)\left(h+\beta(1-\frac{h}{t_k})\right)\nabla^2 f(X(t_k))\dot{X}(t_k)\\
&-\left(1+\frac{\alpha h}{2t_k}\right)\frac{1}{2}\beta h\left(1-\frac{h}{t_k+\frac{\alpha h}{2t_k}}\right)\frac{d^2}{dt^2}(\nabla f\circ X)(t_k)) + \mathcal{O}(h^2) = 0.
\end{aligned}$$

Keeping only the terms of order less than or equal to 1 we obtain

$$\begin{aligned}
&\ddot{X}(t_k) + \frac{\alpha}{t_k}\dot{X}(t_k) + \left(1+\frac{\alpha h+2\beta}{2t_k}\right)\nabla f(X(t_k))\\
&+\left(h+\beta(1-\frac{(\alpha-2)h}{2t_k})\right)\nabla^2 f(X(t_k))\dot{X}(t_k) - \frac{\beta h}{2}\frac{d^2}{dt^2}(\nabla f\circ X)(t_k)) + \mathcal{O}(h^2) = 0.
\end{aligned}$$

We thus obtain the claimed inertial dynamic with Hessian driven damping.

$$\begin{aligned}
&\ddot{X}(t) + \frac{\alpha}{t}\dot{X}(t) + \left(1+\frac{\alpha h+2\beta}{2t}\right)\nabla f(X(t))\\
&+\left(h+\beta(1-\frac{(\alpha-2)h}{2t})\right)\nabla^2 f(X(t))\dot{X}(t)\\
&-\frac{\beta h}{2}\left(\nabla^3 f(X(t))\big(\dot{X}(t),\dot{X}(t)\big) + \nabla^2 f(X(t))\ddot{X}(t)\right) = 0.
\end{aligned}$$

When $\beta = 0$ we recover the high-resolution ODE of (NAG) of order 1

$$\ddot{X}(t) + \frac{\alpha}{t}\dot{X}(t) + \left(1 + \frac{\alpha h}{2t}\right)\nabla f(X(t)) + h\nabla^2 f(X(t))\dot{X}(t) = 0.$$

□

Remark 1. As expected, the correcting term associated with Hessian-driven damping plays a crucial role in the above dynamic through its coefficient $\beta > 0$. Compared to the first-order high-resolution ODE of (NAG), there are significant differences in all the terms that involve the gradient of f. Additionally, we observe that f also contributes through its third-order tensor $\nabla^3 f$.

5 Numerical illustrations

5.1 (RAG) versus (NAG) versus (IGAHD)

We start with a comparison between the three algorithms (RAG), (NAG) and (IGAHD) using the performance profiles on a set of 50 test convex quadratic problems. We extend the comparison of the three algorithms on a non-quadratic convex function, namely the Log-Sum-Exp function which is very popular in machine learning for instance. This non-quadratic function has the advantage of satisfying the classical assumptions commonly used in convergence results, such as the convexity and the Lipschitz continuity of its gradient.

Example 1. Let us perform some numerical tests to compare the three algorithms (RAG), (NAG) and (IGAHD). The functions $f : \mathbb{R}^n \to \mathbb{R}$ are quadratic of the form $f(x) = \frac{1}{2}\|Ax - b\|_2^2$, $A \in \mathbb{R}^{m\times n}$ (with $m \leq n$) and $b \in \mathbb{R}^m$. The matrices A in our set of tests come from the SuiteSparse Matrix Collection[1]. We have chosen a set P of 50 different test problems with matrices $A \in \mathbb{R}^{m\times n}$ sizes ranging from $m = 9$ to $m = 67\,748$ for rows and from $n = 9$ to $n = 216\,350$ for columns. All matrices are selected such that $A^{\mathrm{T}}A$ is singular (or close to be numerically singular) so that the objective-function f to be minimized is convex. The vectors b are randomly selected according to a normal Gaussian distribution. We use the *performance profiles* developed by Dolan-Moré [24] to compare the three methods. For a given performance measure, the performance profile of a solver gives for each $\tau \geq 0$, the proportion of test problems amont the P instances on which the solver has a performance measure within a factor τ of the best possible ratio. For more details, we refer to [24]. We choose three performance measures: the number of iterations, the CPU-time and the residuals $R(x) = \|A^{\mathrm{T}}(b - Ax)\|_2$ (or the gradient norm), where x is the numerical solution found by each algorithm. The value of $\alpha \geq 3$ is fixed and taken equal to 5. We use the same starting points and the same stopping criteria *i.e.* either the number of iterations exceeds 10^5 or $\|\nabla f(x_k)\| \leq 10^{-7}$. In this case, a failure is declared if the number of iterations exceeded the maximum number of iterations. If we consider the number of iterations (top left), we observe that (IGAHD) outperforms both (RAG) and (NAG). In fact, for $\tau \leq \frac{1}{2}$, (RAG) and (NAG) have solved less than 45% while (IGAHD) has solved almost 92% of the problems. We also observe that (RAG) and (NAG) have similar performances with a very slight advantage to (RAG). This can be explained by the fact that these two algorithms are numerically close: a gradient and a linear extrapolation for each step. If we consider the CPU-time (top right) as performance measures, (IGAHD) consistently

[1] https://sparse.tamu.edu.

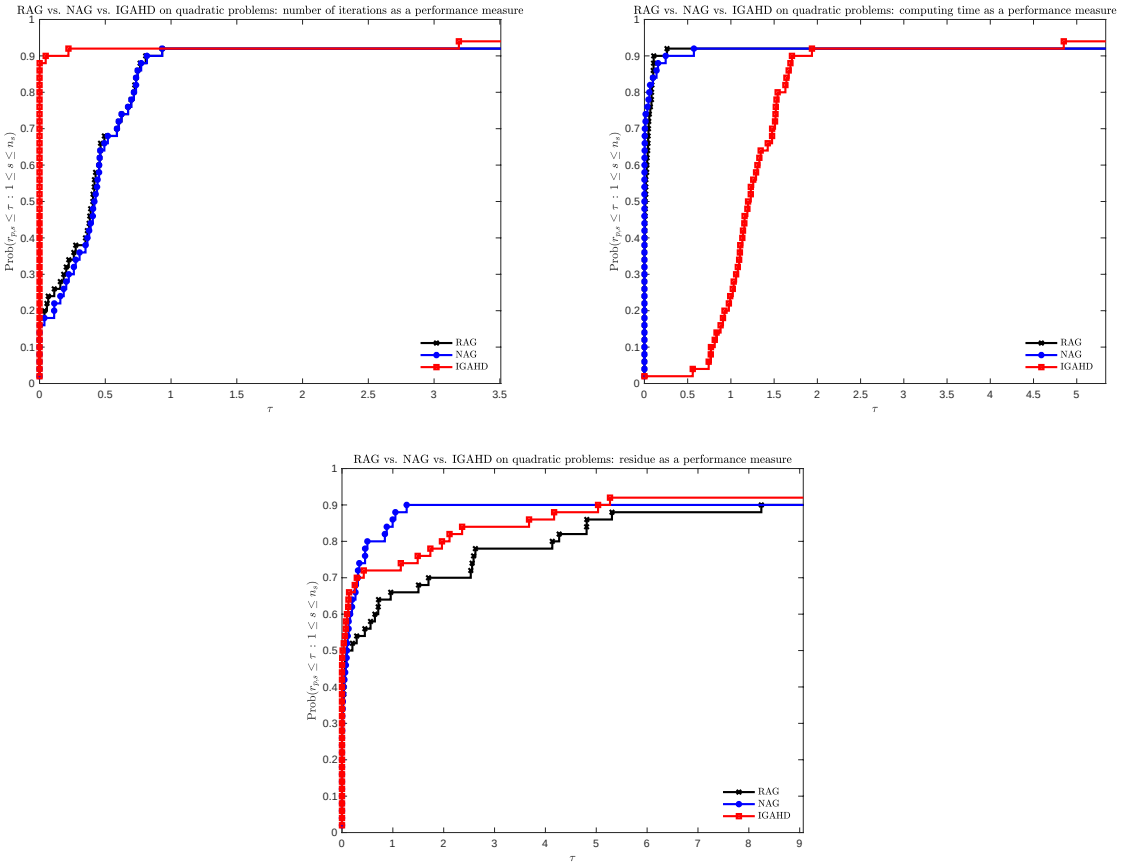

Figure 4: Performance profiles of (RAG), (NAG) and (IGAHD) on quadratic convex functions.

exhibits significantly lower CPU-time efficiency, primarily due to its double invocation of the gradient calculation during each iteration. In contrast, both (RAG) and (NAG), sharing the exact same per-iteration cost, demonstrate similar and notably superior CPU performance compared to (IGAHD). On the other hand, if we consider the residual norm as a performance measure (bottom), we observe that the three algorithms have similar behavior for small τ and that for $\tau \geq \frac{1}{2}$, (NAG) outperforms the two other solvers.
Therefore, we can conclude that, under the same stopping criteria and with the same initial points, (IGAHD) outperforms both (RAG) and (NAG) in terms of the number of iterations required for convergence. However, when considering CPU-time as a performance measure, (IGAHD) consistently exhibits lower efficiency due to its double gradient calculation per iteration. Furthermore, the three algorithms show comparable behavior for small τ values in terms of the residual norms, but (NAG) demonstrates a slight advantage for larger τ values. Based on the analysis of performance profiles on a set of 50 test convex quadratic functions, it becomes apparent that these three numerical methods exhibit trade-offs and distinct variations in their performance characteristics.

Example 2. In this example, we compare the three algorithms (RAG), (NAG) and (IGAHD) using the performance profiles on the following "Log-Sum-Exp" function $f : \mathbb{R}^n \to \mathbb{R},\ x = (x_1, \ldots, x_n) \mapsto$

$f(x)$ defined by

$$f(x) = \rho \log\Big(\sum_{i=1}^{m} \exp\Big(\frac{\langle a_i,\, x\rangle - b_i}{\rho}\Big)\Big),$$

where $\rho > 0$, $a_i \in \mathbb{R}^n$, $i = 1, \ldots, m$ and $b = (b_1, \ldots, b_m) \in \mathbb{R}^m$ are some given data. We set $A = [a_1 \quad a_2 \quad \ldots \quad a_m]^{\mathrm{T}} \in \mathbb{R}^{m\times n}$, $R(x) = \frac{1}{\rho}(Ax - b)$, $Z(x) = \exp(R(x))$ (exponential to be understood pointwise) and $z(x) = \langle 1_n,\, Z(x)\rangle$. We then have

$$\nabla f(x) = \frac{1}{z(x)} A^{\mathrm{T}} Z(x) \quad \text{and} \quad \nabla^2 f(x) = \frac{1}{\rho} A^{\mathrm{T}} \left(\frac{1}{z(x)} \mathrm{diag}(Z(x)) - \frac{1}{z(x)^2} Z(x) Z(x)^{\mathrm{T}}\right) A.$$

It is easy to show that the Hessian matrix $\nabla^2 f(x)$ is positive semi-definite and that ∇f is L-Lipschitz with $L = \frac{2}{\rho}\|A\|^2$. We set $m = 6n$ in the numerical simulations. For the performance profiles in Figure 5, we generate randomly 50 matrices A such that $5 \leq n \leq 100$ with $m = 6n$ and 50 vectors $b \in \mathbb{R}^n$. The values of the parameter ρ are taken randomly in the interval $[1, 50]$. Figure 5 shows that, when considering the number of iterations as a performance metric, (IGAHD) demonstrates superior performance compared to both (RAG) and (NAG) for this specific example. Additionally, we observe that (RAG) exhibits slightly better performance than (NAG). As for the quadratic case, the comparison based on CPU-time as a performance metric showed that both (RAG) and (NAG) outperformed (IGAHD), with a slight advantage observed for (RAG). This can be attributed to the fact that (IGAHD) necessitates two gradient calculations per iteration. In contrast, when considering gradient norms as a performance measure (bottom), all three algorithms exhibit similar behavior for small τ values. However, for $\tau \geq 0.1$, (IGAHD) outperforms the other two solvers, demonstrating superior performance in this specific example.

5.2 Comparison of the continuous ODEs of (RAG), (NAG) and (IGAHD)

In this subsection, we conduct some numerical experiments to compare the continuous low- and high-resolution ordinary differential equations (ODEs) associated with (RAG), (NAG), and (IGAHD). We will solve all involved continuous dynamics using adapted standard Runge-Kutta integrators. For further investigation of gradient-based optimization methods and their relationship with the discretization of second-order ODEs using the Runge-Kutta method, we refer to [42] and the references cited therein.

Example 3. Let us consider the following simple example in $\mathbb{R}^2$ where f is a convex quadratic function. Specifically, we take $f(x_1, x_2) = ax_1^2 + bx_2^2$ with $a = 0.02$ and $b = 0.005$, with the initial condition $x_0 = (2, 2)$ and $x_1 = (1, 1)$. We note that this function was studied in [38, Figure 1] and also in [36, Figure 2] in the context of continuous ODEs associated with NAG. We illustrate Theorem 2, Theorem 4, and Theorem 7 by comparing the low-resolution ODE (AVD_α), the high-resolution ODE of order 1 in h of (NAG), (RAG), and (IGAHD), and the super-resolution ODE of order 2 in h of NAG and RAG. We note that for this quadratic function, the ODEs (2) and (9) are the same for NAG and RAG since the third-order derivatives of f are null in this case.
Figure 6 shows the trajectories of the low-resolution (AVD_α), the high-resolution of order h of (NAG), (RAG), and (IGAHD), and the super high-resolution of order h^2 of NAG and RAG (left part). The corresponding error values on a logarithmic scale are depicted in Figure 6 (right part). We plotted three trajectories and the corresponding error values for $s = 0.2$, $s = 0.1$, and $s = 0.01$. We observe that when the parameter s approaches 0, all trajectories converge to the solution of

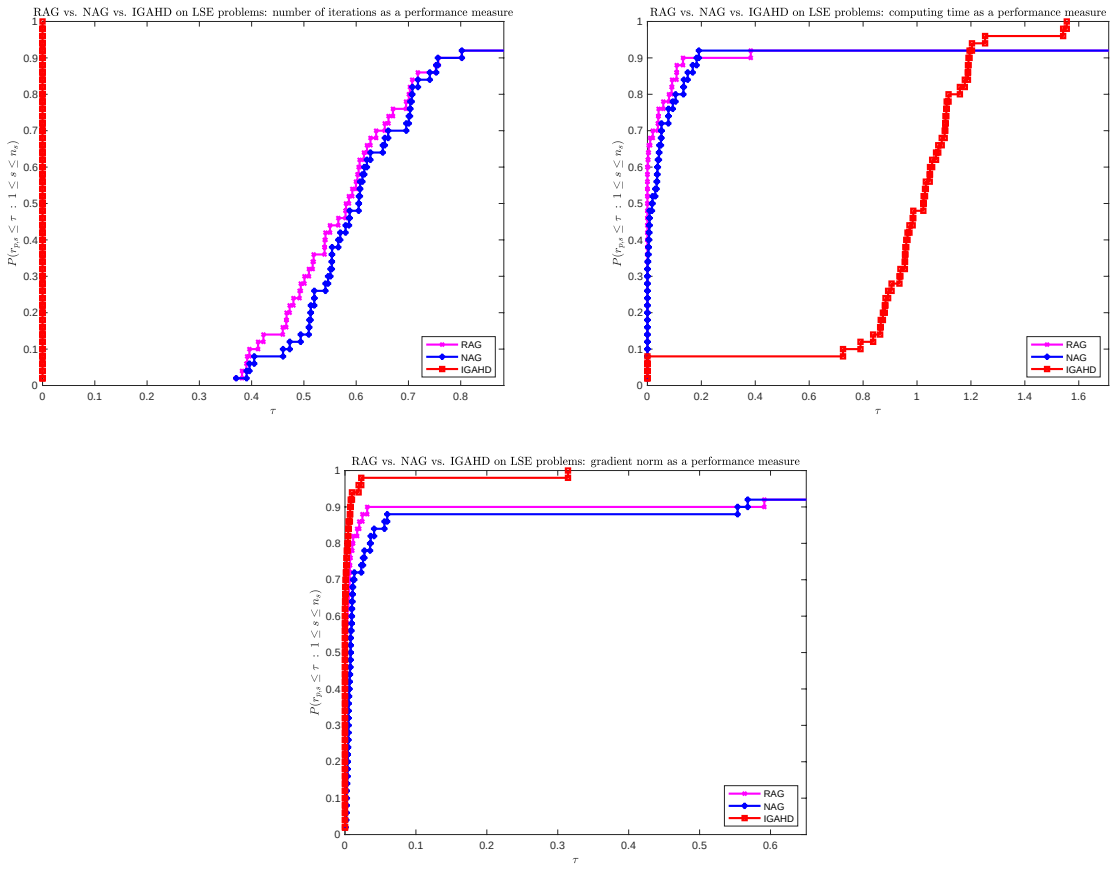

Figure 5: Performance profiles of RAG, NAG and (IGAHD) on 50 problems with Log-Sum-Exp function (Example 2).

(AVD$_\alpha$). This corresponds to the fact that when $s = 0$, the dynamics (3), (2), and (9) coincide with (AVD$_\alpha$).

Example 4. To establish a comparison between the super-resolution ODE of NAG and RAG, we consider the following convex and non-quadratic function $f : \mathbb{R}^2 \to \mathbb{R},\ x = (x_1, x_2) \mapsto f(x_1, x_2) = ax_1^4 + bx_2^4$, with $a,\ b \geq 0$. In order to compute the third derivative in the ODEs (2) and (9), we use the following observation

$$\nabla^3 f(x(t))\big(\dot{x}(t), \dot{x}(t)\big) = \frac{d}{dt}\big[\nabla^2 f(x(t)\dot{x}(t)\big] - \nabla^2 f(x(t))\ddot{x}(t),$$

which gives in our case

$$\nabla^3 f(x(t))\big(\dot{x}(t), \dot{x}(t)\big) = \Big(24ax_1(t)\dot{x}_1^2(t), 24bx_2(t)\dot{x}_2^2(t)\Big).$$

In Figure 7, we have depicted two trajectories of the super-resolution ODE of order h^2 for NAG and RAG, respectively, under distinct initial conditions. We have considered the parameter values

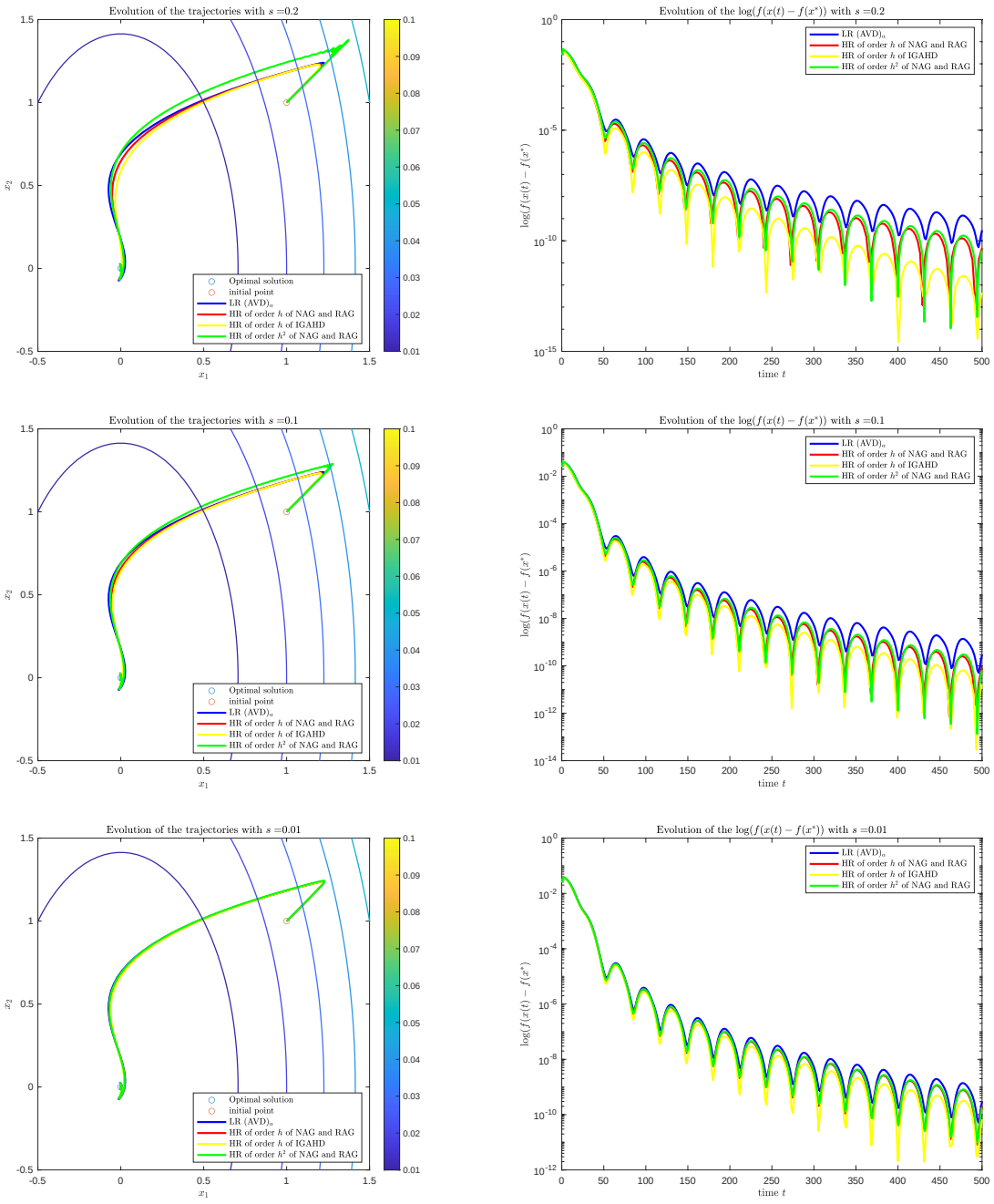

Figure 6: Trajectories of the low-resolution (LR), the high-resolution (HR) of order h and the high-resolution of order h^2 of NAG and RAG for $s = 0.2$, $s = 0.1$ and $s = 0.01$ (left). The corresponding errors-values $t \mapsto \log(f(x(t) - f(x^*))$ (right) (Example 3).

$a = b = 1$ and $s = 0.1$. It is evident from the plot that these two trajectories exhibit stark dissimilarities, indicating the disparate nature of the two dynamics when applied to non-quadratic

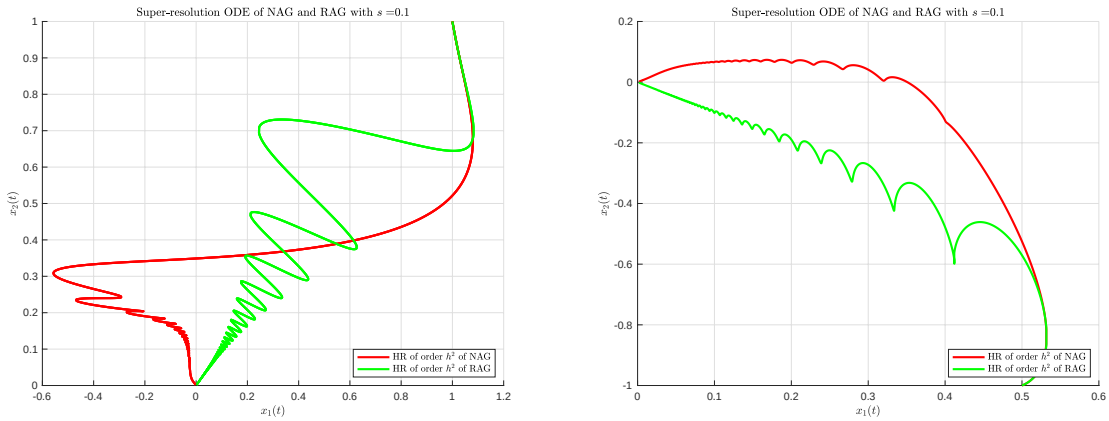

Figure 7: Trajectories of the super-resolution of order h^2 of NAG and RAG with $s = 0.1$ for different initial conditions (Example 4).

functions. Naturally, when the parameter s assumes a small value, the two dynamics will converge to the low-resolution ODE (AVD_α) and exhibit similar behavior.

6 Discussion of the differences between (NAG), (RAG) and (IGAHD)

6.1 (NAG) and (RAG) as different algorithms

- The claims of Theorem 2 and Theorem 4 clearly show that the two algorithms (NAG) and (RAG) are different. The corresponding super-resolution ODE's reveal a difference between them: the second-order remainder is different, and this differences lies precisely in the third-order tensor term $\nabla^3 f$ which enters the (NAG) and (RAG) super-resolution ODE's with opposite signs (sign plus for (NAG) and minus for (RAG)). We note that on quadratic functions, these third-order derivatives vanish and both high-resolution ODE's surrogates of (NAG) and (RAG) coincide. Thus the difference between the two methods can only be observed on functions such that $\nabla^3 f$ is not identically zero.

- The high-resolution ODEs for (NAG) and (RAG) are expressed in terms of two different variables: x for (NAG) and y for (RAG). It's worth noting that to obtain the high-resolution ODE for (NAG), an additional Taylor expansion on the gradient is required to bring out the Hessian. In contrast, the Hessian appears explicitly in the discretization of the ODE (9) associated with (RAG).

- The remarkable success of (NAG) can be attributed to its ability to adapt well to convex optimization problems with an additive "smooth + not smooth" structure. This makes it similar to the FISTA inertial proximal gradient algorithm proposed by Beck and Teboulle [20]. In contrast, finding the correct form of the proximal gradient inertial algorithm for (RAG) is still an ongoing research topic.

The table below provides a summary of the relationships between the three accelerated gradient algorithms, providing a bird's-eye view. It serves as a complement to the table presented in [14].

Algorithm	(RAG)	(NAG)	(IGAHD)
Low-resolution ODE	(AVD$_\alpha$)	(AVD$_\alpha$)	(AVD$_\alpha$)
High-resolution ODE	Hessian driven damping	Hessian driven damping	Hessian driven damping
Super-resolution ODE	h^2 correcting term	h^2 correcting term	Different h^2 correcting term
Fast convergence of the gradients	Yes	Yes	Yes
Convergence rate, $\alpha > 3$	$f(y_k) - \min_{\mathcal{H}} f = o\left(\frac{1}{k^2}\right)$	$f(x_k) - \min_{\mathcal{H}} f = o\left(\frac{1}{k^2}\right)$	$f(y_k) - \min_{\mathcal{H}} f = o\left(\frac{1}{k^2}\right)$
Convergence of iterates, $\alpha > 3$	Yes	Yes	Yes
Composite optimization, $\alpha > 3$	Open question	FISTA	Open question

6.2 Comparison of (IGAHD) with (NAG) and (RAG)

- The high resolution of (IGAHD) explains why it outperforms the other two algorithms, namely (NAG) and (RAG). This is particularly evident when the function f is ill-conditioned. When the Lipschitz constant L of ∇f is large, it is necessary to choose a step size s less than or equal to $\frac{2}{L}$ to ensure convergence of the algorithms (NAG) and (RAG). However, this results in a very small coefficient $\sqrt{s}$ for the Hessian-driven damping term that appears in the high resolution of both algorithms, which fails to effectively dampen the oscillations. In contrast, the coefficient β of the Hessian-driven damping term that appears in the high resolution of (IGAHD) remains fixed, and therefore has a significant effect on attenuating the oscillations (see Figure 8 for an illustration on a ill-conditioned quadratic problem).

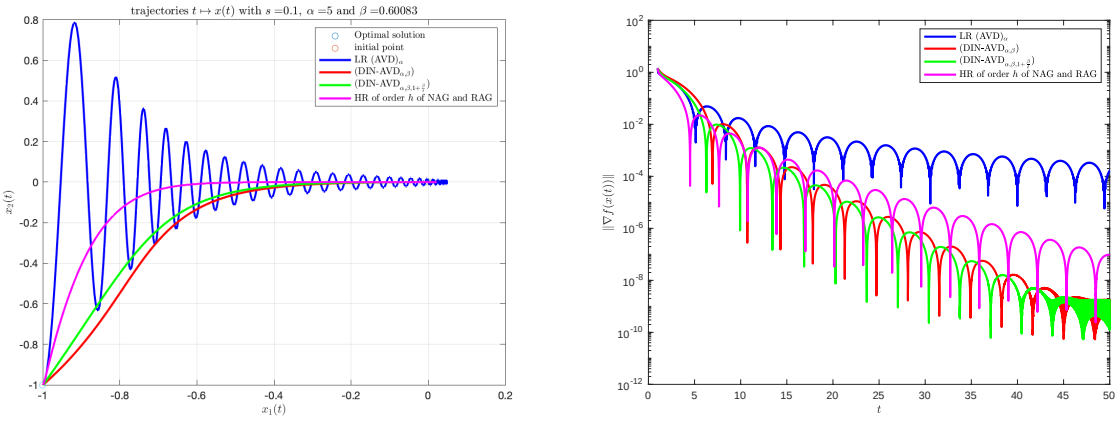

Figure 8: Evolution of the trajectories of the dynamics (AVD$_\alpha$), (DIN-AVD$_{\alpha,\beta}$), (DIN-AVD$_{\alpha,\beta,1+\frac{\beta}{t}}$) and high-resolution ODE of NAG-RAG (left) and $\|\nabla f(x(\cdot))\|$ (right) on an ill-conditioned quadratic problem in $\mathbb{R}^2$.

- The algorithms (NAG) and (IGAHD) are both derived using the same discretization procedure, which involves applying an implicit scheme followed by the replacement of the proximal operator with a gradient. However, they originate from two distinct dynamical systems: (AVD$_\alpha$) for (NAG), and (DIN-AVD) for (IGAHD). Since (DIN-AVD) satisfies fast convergence of the gradients while (AVD$_\alpha$) does not, it is not surprising to observe a significant

difference between the corresponding algorithms. It should be noted that for this argument to hold, the same discretization procedure must be used. Indeed, recent work [18] has demonstrated the superiority of (DIN-AVD) over (AVD_α) in the strongly convex case through complexity analysis.

7 Conclusion, Perspective

In this paper we have highlighted the similarities as well as the differences between the three algorithms. Differentiating them is important, because even if they are close, when they intervene as basic blocks of splitting algorithms (proximal gradient method, primal dual methods, ADMM), they can give rise to clearly different algorithms. Based on the super-resolution ODE thechnique we have been able to show that (NAG) and (RAG) are different algorithms. Based on the high-resolution ODE thechnique we have been able to explain why IGADH provide a better Hessian driven damping than the two others, especially for ill conditioned optimization problems. We have also conducted numerical experiments to support the theoretical part. Several questions remain open and require further investigations. The additively structure "smooth+nonsmooth" convex problems are very important in applications. Finding the correct form of the proximal gradient inertial algorithm for (IGAHD) and (RAG) is an open question. The extension to nonsmooth convex optimization by using differential inclusion involving the subdifferential of the potential or a maximally monotone operator (in the nonpotential case) is also of great interest. This is beyond the scope of this manuscript and will be the subject of a future research project.

References

[1] S. Adly, H. Attouch, *Finite convergence of proximal-gradient inertial algorithms combining dry friction with Hessian-driven damping*, SIAM J. Optim., 30(3) (2020), 2134–2162.

[2] S. Adly, H. Attouch, *Finite time stabilization of continuous inertial dynamics combining dry friction with Hessian-driven damping*, Journal of Convex Analysis, 28 (2) (2021).

[3] C.D. Alecsa, S.C.László, T. Pinta, *An extension of the second order dynamical system that models Nesterovs convex gradient method*, Appl. Math. Optim. (2020). https://doi.org/10.1007/s00245-020-09692-1

[4] F. Álvarez, H. Attouch, J. Bolte, P. Redont, *A second-order gradient-like dissipative dynamical system with Hessian-driven damping. Application to optimization and mechanics*, J. Math. Pures Appl., 81(8) (2002), 747–779.

[5] V. Apidopoulos, J.-F. Aujol, Ch. Dossal, *Convergence rate of inertial Forward-Backward algorithm beyond Nesterov's rule*, Math. Program., 180 (2020), 137–156.

[6] V. Apidopoulos, J.-F. Aujol, Ch. Dossal, The differential inclusion modeling the FISTA algorithm and optimality of convergence rate in the case $b \leq 3$, SIAM J. Optim., 28(1) (2018), 551—574.

[7] H. Attouch, A. Cabot, *Asymptotic stabilization of inertial gradient dynamics with time-dependent viscosity*, J. Differential Equations, 263 (2017), pp. 5412-5458.

[8] H. Attouch, A. Cabot, *Convergence rates of inertial forward-backward algorithms*, SIAM J. Optim., 28 (1) (2018), 849–874.

[9] H. Attouch, A. Cabot, Z. Chbani, H. Riahi, *Rate of convergence of inertial gradient dynamics with time-dependent viscous damping coefficient*, Evolution Equations and Control Theory, 7 (2018), No. 3, pp. 353–371

[10] H. Attouch, Z. Chbani, J. Fadili, H. Riahi, *First-order optimization algorithms via inertial systems with Hessian driven damping*, Math. Program., 193 (2022), 113–155.

[11] H. Attouch, Z. Chbani, J. Fadili, H. Riahi, *Convergence of iterates for first-order optimization algorithms with inertia and Hessian driven damping*, (2021), Optimization, arXiv:2107.05943v1 [math.OC] Jul 2021

[12] H. Attouch, Z. Chbani, J. Peypouquet, P. Redont, *Fast convergence of inertial dynamics and algorithms with asymptotic vanishing viscosity*, Math. Program. Ser. B 168 (2018), 123–175.

[13] H. Attouch, Z. Chbani, H. Riahi, *Rate of convergence of the Nesterov accelerated gradient method in the subcritical case* $\alpha \leq 3$, ESAIM-COCV, 25 (2019), Article Number 2, https://doi.org/10.1051/cocv/2017083

[14] H. Attouch, J. Fadili, *From the Ravine Method to the Nesterov Method and Vice Versa: A Dynamical System Perspective*, SIAM J. Optim. 32(3) (2022), 10.1137/22M1474357

[15] H. Attouch, J. Peypouquet, *The rate of convergence of Nesterov's accelerated forward-backward method is actually faster than* $1/k^2$, SIAM J. Optim., 26(3) (2016), 1824–1834.

[16] H. Attouch, J. Peypouquet, P. Redont, *A dynamical approach to an inertial forward-backward algorithm for convex minimization*, SIAM J. Optim., 24 (2014), No. 1, pp. 232–256.

[17] H. Attouch, J. Peypouquet, P. Redont, *Fast convex minimization via inertial dynamics with Hessian driven damping*, J. Differential Equations, 261 (2016), 5734–5783.

[18] J.-F. Aujol, C. Dossal, V. H. Hoang, H. Labarriere *Fast convergence of inertial dynamics with Hessian-driven damping under geometry assumptions*, (2022), arXiv:2206.06853v3 [math.OC].

[19] J.-F. Aujol, C. Dossal, A. Rondepierre, *Optimal convergence rates for Nesterov acceleration*, SIAM Journal on Optimization, 29 (4) (2019), pp. 3131–3153.

[20] A. Beck, M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), No. 1, pp. 183–202.

[21] A. Cabot, H. Engler, S. Gadat, *On the long time behavior of second order differential equations with asymptotically small dissipation*, Transactions of the American Mathematical Society, 361 (2009), pp. 5983–6017.

[22] A. Cabot, H. Engler, S. Gadat, *Second order differential equations with asymptotically small dissipation and piecewise flat potentials*, Electronic Journal of Differential Equations, 17 (2009), pp. 33–38.

[23] A. Chambolle, Ch. Dossal, *On the convergence of the iterates of the Fast Iterative Shrinkage Thresholding Algorithm*, J. Opt. Theory and Appl., 166 (2015), 968–982.

[24] E. D. Dolan, J. J. Moré, *Benchmarking Optimization Software with Performance Profiles*, Math. Program., 91 (2002), pp. 201–213.

[25] I.M. Gelfand, M. Tsetlin, *Printszip nelokalnogo poiska v sistemah avtomatich*, Optimizatsii, Dokl. AN SSSR, 137 (1961), pp. 295–298 (in Russian).

[26] D. Kim, J.A. Fessler, *Optimized first-order methods for smooth convex minimization*, Math. Program. 159(1) (2016), 81–107.

[27] J. Liang, J. Fadili, G. Peyré, *Local linear convergence of forward-backward under partial smoothness*, Advances in Neural Information Processing Systems, 2014, pp. 1970–1978.

[28] R. May, *Asymptotic for a second-order evolution equation with convex potential and vanishing damping term*, Turkish Journal of Math., 41(3) (2017), 681–685.

[29] M. Muehlebach, M. I. Jordan, *A Dynamical Systems Perspective on Nesterov Acceleration*, Proceedings of the International Conference on Machine Learning, 2019, https://arxiv.org/abs/1905.07436

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

[31] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*, volume 87 of Applied Optimization. Kluwer Academic Publishers, Boston, MA, 2004.

[32] Y. Nesterov, *How to Make the Gradients Small*, Discussion Column, Optima, Mathematical Optimization Society Newsletter, 88 (2012), 10–11.

[33] J. Pedlosky, *Geophysical Fluid Dynamics*, Springer Science and Business Media, Berlin (2013).

[34] B.T. Polyak, *Accelerated gradient methods revisited*, Workshop Variational Analysis and Applications, August 28-September 5, 2018, Erice.

[35] W. Siegel, *Accelerated first-order methods: Differential equations and Lyapunov functions*, arXiv:1903.05671v1 [math.OC], 2019.

[36] B. Shi, S.S. Du, M. I. Jordan, W. J. Su, *Understanding the acceleration phenomenon via high-resolution differential equations*, Math. Program. (2022) 195:79-148.

[37] B. Shi, S.S. Du, W. J. Su, M. I. Jordan, *Acceleration via symplectic discretization of high-resolution differential equations*. NeurIPS (2019).

[38] W. J. Su, S. Boyd, E. J. Candès, *A differential equation for modeling Nesterov's accelerated gradient method: theory and insights*. Neural Information Processing Systems 27 (2014), 2510–2518.

[39] Suvrit Sra, *Optimization for Machine Learning: Subgradient method; Accelerated gradient*, Massachusetts Institute of Technology, March 2021.

[40] S. Villa, S. Salzo, L. Baldassarres, A. Verri, *Accelerated and inexact forward-backward*, SIAM J. Optim., 23 (2013), No. 3, pp. 1607–1633 .

[41] D. Wu. β-high-resolution ODE and phase transition between NAG-SC and Heavy ball method, arXiv:2004.03121v1 (2020).

[42] J. Zhang, A. Mokhtari, S. Sra, A. Jadbabaie, *Direct Runge-Kutta Discretization Achieves Acceleration*, https://arxiv.org/abs/1805.00521.