Conformal Symplectic and Relativistic Optimization

Guilherme França,[*,1,2] **Jeremias Sulam**,[2] **Daniel P. Robinson**,[3] **and René Vidal**[2]

[1]*Division of Computer Science, University of California, Berkeley, CA, USA*
[2]*Mathematical Institute for Data Science, Johns Hopkins University, MD, USA*
[3]*Industrial and Systems Engineering, Lehigh University, PA, USA*

Abstract

Arguably, the two most popular accelerated or momentum-based optimization methods in machine learning are Nesterov's accelerated gradient and Polyaks's heavy ball, both corresponding to different discretizations of a particular second order differential equation with friction. Such connections with continuous-time dynamical systems have been instrumental in demystifying acceleration phenomena in optimization. Here we study structure-preserving discretizations for a certain class of dissipative (conformal) Hamiltonian systems, allowing us to analyze the symplectic structure of both Nesterov and heavy ball, besides providing several new insights into these methods. Moreover, we propose a new algorithm based on a dissipative relativistic system that normalizes the momentum and may result in more stable/faster optimization. Importantly, such a method generalizes both Nesterov and heavy ball, each being recovered as distinct limiting cases, and has potential advantages at no additional cost.

[*] guifranca@gmail.com
December 2020

Contents

1 Introduction

Gradient based optimization methods are ubiquitous in machine learning since they only require first order information on the objective function. This makes them computationally efficient. However, vanilla gradient descent can be slow. Alternatively, *accelerated gradient methods*, whose construction can be traced back to Polyak [1] and Nesterov [2], became popular due to their ability to achieve best worst-case complexity bounds. The heavy ball method, also known as *classical momentum* (CM) method, is given by

$$v_{k+1} = \mu v_k - \epsilon \nabla f(x_k), \qquad x_{k+1} = x_k + v_{k+1}, \tag{1.1}$$

where $k = 0, 1, \ldots$ is the iteration number, $\mu \in (0, 1)$ is the momentum factor, $\epsilon > 0$ is the learning rate, and $f : \mathbb{R}^n \to \mathbb{R}$ is the function being minimized. Similarly, *Nesterov's*

accelerated gradient (NAG) can be found in the form

$$v_{k+1} = \mu v_k - \epsilon \nabla f(x_k + \mu v_k), \qquad x_{k+1} = x_k + v_{k+1}. \tag{1.2}$$

Both methods have a long history in optimization and machine learning [3]. They are also the basis for the construction of other methods, such as adaptive ones that additionally include some gradient normalization [4–7].

In discrete-time optimization the "acceleration phenomena" are considered counterintuitive. By this we mean a mechanism by which an algorithm can be accelerated, i.e. have a faster convergence; for instance, it is known that gradient descent converges at a rate of $O(1/k)$ for convex functions, while NAG converges at a rate $O(1/k^2)$, which is optimal in the sense of worst-case complexity. A complete understanding of why NAG is able to achieve such an improved rate is considered by many experts an important open problem, and currently there is no guiding principle to construct accelerated algorithms. A promising direction to understand this has been emerging in connection with continuous-time dynamical systems [8–18] where many of these difficulties disappear or have an intuitive explanation. Since one is free to discretize a continuous-time system in many different ways, it is only natural to ask which discretization strategies would be most suitable for optimization? Such a question is unlikely to have a simple answer, and may be problem dependent. Unfortunately, typical discretizations are also known to introduce spurious artifacts and do not reproduce the most important properties of the continuous-time system [19]. Nevertheless, a special class of discretizations in the physics literature known as *symplectic integrators* [19–22] are to be preferable whenever considering the special class of *conservative Hamiltonian systems*.

More relevant to optimization is a class of *dissipative* systems known as *conformal Hamiltonian systems* [23]. Recently, results from symplectic integrators were extended to this case and such methods are called *conformal symplectic integrators* [18,24]. Conformal symplectic methods tend to have long time stability because the numerical trajectories remain in the same conformal symplectic manifold as the original system [18]. Importantly, these methods do not change the phase portrait of the system, i.e. the stability of critical points is preserved. Although symplectic techniques have had great success in several areas of physics and Monte Carlo methods, only recently they started to be considered in optimization [14, 18] and are still mostly unexplored in this context. Very recently a great progress has been made [18] by showing that such an approach is able to preserve the continuous-time rates of convergence up to a controlled error [18].

In this paper, we *relate conformal symplectic integrators to optimization* and provide important insights into CM (1.1) and NAG (1.2). We prove that CM is a first order accurate conformal symplectic integrator. On the other hand, we show that NAG is also first order accurate, but not conformal symplectic since it introduces some spurious dissipation—or excitation. However, it does so in an interesting way that depends on the Hessian $\nabla^2 f$; the symplectic form contracts in a Hessian dependent manner and so do phase space volumes. This is an effect of higher order but can influence the behaviour of the algorithm. We also derive *modified equations* and *shadow Hamiltonians* for both CM and NAG. Moreover, we indicate a tradeoff between stability, symplecticness, and such an spurious contraction, indicating advantages in structure-preserving discretizations for optimization.

Algorithm 1 RGD method for minimizing a smooth function $f(x)$. In practice, we recommend setting $\alpha = 1$ which results in a conformal symplectic method.

Require: Initial state (x_0, v_0) and parameters $\epsilon > 0$, $\delta > 0$, $\mu \in (0, 1)$, $\alpha \in [0, 1]$

for $k = 0, 1, \ldots$ **do**

$$x_{k+1/2} \leftarrow x_k + \sqrt{\mu}\,(\mu\delta\|v_k\|^2 + 1)^{-1/2}\, v_k$$
$$v_{k+1/2} \leftarrow \sqrt{\mu} v_k - \epsilon\nabla f(x_{k+1/2})$$
$$x_{k+1} \leftarrow \alpha x_{k+1/2} + (1-\alpha)x_k + \left(\delta\|v_{k+1/2}\|^2 + 1\right)^{-1/2} v_{k+1/2}$$
$$v_{k+1} \leftarrow \sqrt{\mu}\, v_{k+1/2}$$

end for

Optimization can be challenging in a landscape with large gradients, e.g. for a function with fast growing tails. The only way to control divergences in methods such as (1.1) and (1.2) is to make the step size very small, but then the algorithm becomes slow. One approach to this issue is to introduce a suitable normalization of the gradient. Here we propose an alternative approach motivated by *special relativity* in physics. The reason is that in special relativity there is a limiting speed, i.e. the speed of light. Thus, by discretizing a *dissipative relativistic system*, we obtain an algorithm that incorporates this effect and may result in more stable optimization in settings with large gradients. Specifically, we introduce Algorithm 1. Besides the momentum factor μ and the learning rate ϵ—also present in (1.1) and (1.2)—the above *relativistic gradient descent* (RGD) method has the additional parameters $\delta \geq 0$ and $0 \leq \alpha \leq 1$ which brings some interesting properties:

- When $\delta = 0$ and $\alpha = 0$, RGD recovers NAG (1.2). When $\delta = 0$ and $\alpha = 1$, RGD becomes a second order accurate version of CM (1.1), which has a close behavior but an improved stability. Thus, RGD can interpolate between these two methods. Moreover, RGD has the same computational cost as CM or NAG. These facts imply that RGD is at least as efficient as CM and NAG if appropriately tuned.

- Let $y_k \equiv \alpha x_{k+1/2} + (1-\alpha)x_k$. The last update in Algorithm 1 implies $\|x_{k+1} - y_k\| \leq 1/\delta$. Thus, with $\delta > 0$, RGD is globally bounded regardless how large $\|\nabla f\|$ might be; this is in contrast with CM and NAG where $\delta = 0$, i.e. $\|x_{k+1} - y_k\| \leq \infty$. The square root factor in Algorithm 1 has a "relativistic origin" and its strength is controlled by δ. For this reason, RGD may be more stable than CM and NAG, preventing divergences in settings of large gradients; see Fig. 1 in Section 6 and the plots in Appendix B.

- As we will show, $\alpha = 1$ implies that RGD is *conformal symplectic*, whereas $\alpha = 0$ implies a spurious Hessian driven damping similarly found in NAG. Thus, RGD has the flexibility of being "dissipative-preserving" or introducing some "spurious contraction." However, based on theoretical arguments and empirical evidence, we advocate for the choice $\alpha = 1$.[1]

[1]The only reason for introducing the extra parameter $0 \leq \alpha \leq 1$ into Algorithm 1 is to actually let the experiments decide whether $\alpha = 1$ (symplectic) or $\alpha < 1$ (non-symplectic) is desirable or not.

Let us mention a few related works. Applications of symplectic integrators in optimization was first considered in [14]—although this is different than the conformal symplectic case explored here. Recently, the benefits of symplectic methods in optimization started to be indicated [25]. Actually, even more recently, a generalization of symplectic integrators to a general class of dissipative Hamiltonian systems was proposed [18], with theoretical results ensuring that such discretizations are "rate-matching" up to a negligible error; this construction is general and contains the conformal case considered here as a particular case. Relativistic systems are obviously an elementary topic in physics but—with some modifications—the relativistic kinetic energy was considered in Monte Carlo methods [26,27] and also briefly in [28]. Finally, we stress that Algorithm 1 is a completely new method in the literature, generalizing perhaps the two most popular existing accelerated methods, namely CM and NAG, and also has the ability to be conformal symplectic besides being adaptive in the momentum which may help controlling divergences. We also provide several new insights into CM and NAG in Section 4.3 and Section 6 which may be of independent interest.

2 Conformal Hamiltonian systems

We start by introducing the basics of conformal Hamiltonian systems and focus on their intrinsic symplectic geometry; we refer to [23] for details. The state of the system is described by a point on phase space, $(x, p) \in \mathbb{R}^{2n}$, where $x = x(t)$ is the generalized coordinates and $p = p(t)$ its conjugate momentum, with $t \in \mathbb{R}$ being the time. The system is completely specified by a Hamiltonian function $H : \mathbb{R}^{2n} \to \mathbb{R}$ and required to obey a modified form of Hamilton's equations:

$$\dot{x} = \nabla_p H(x,p), \qquad \dot{p} = -\nabla_x H(x,p) - \gamma p. \tag{2.1}$$

Here $\dot{x} \equiv \frac{dx}{dt}$, $\dot{p} \equiv \frac{dp}{dt}$, and $\gamma > 0$ is a damping constant. A classical example is given by

$$H(x,p) = \frac{\|p\|^2}{2m} + f(x) \tag{2.2}$$

where $m > 0$ is the mass of a particle subject to a potential f. The Hamiltonian is the energy of the system and upon taking its time derivative one finds $\dot{H} = -\gamma\|p\|^2 \leq 0$. Thus H is a Lyapunov function and all orbits tend to critical points, which in this case must satisfy $\nabla f(x) = 0$ and $p = 0$. This implies that the system is stable on isolated minimizers of f.[2]

Define

$$z \equiv \begin{bmatrix} x \\ p \end{bmatrix}, \qquad \Omega \equiv \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}, \qquad D \equiv \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}, \tag{2.3}$$

where I is the $n \times n$ identity matrix, to write the equations of motion (2.1) concisely as[3]

$$\dot{z} = \underbrace{\Omega \nabla H(z)}_{C(z)} - \underbrace{\gamma D z}_{D(z)}. \tag{2.4}$$

[2]This can be generalized for any Hamiltonian H that is strongly convex on p with the minimum at $p = 0$.

[3]$C(z)$ and $D(z)$ will be used later on and stand for "conservative" and "dissipative" parts, respectively.

Note that $\Omega\Omega^T = \Omega^T\Omega = I$ and $\Omega^2 = -I$, so that Ω is real, orthogonal and antisymmetric. Let $\xi, \eta \in \mathbb{R}^{2n}$ and define the *symplectic 2-form* $\omega(\xi, \eta) \equiv \xi^T \Omega\, \eta$. It is convenient to use the wedge product representation of this 2-form, namely[4]

$$\omega(\xi, \eta) = (dx \wedge dp)(\xi, \eta). \tag{2.5}$$

We denote $\omega_t \equiv dx(t) \wedge dp(t)$. The equations of motion define a flow $\Phi_t : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$, i.e. $\Phi_t\big(z_0\big) \equiv z(t)$ where $z(0) \equiv z_0$. Let $J_t(z)$ denote the Jacobian of $\Phi_t(z)$. From (2.4) it is not hard to show that (see e.g. [23])

$$J_t^T \Omega J_t = e^{-\gamma t}\Omega \quad \implies \quad \omega_t = e^{-\gamma t}\omega_0. \tag{2.6}$$

Therefore, a conformal Hamiltonian flow Φ_t *contracts the symplectic form exponentially* with respect to the damping coefficient γ. It follows from (2.6) that volumes on phase space shrink as $\mathrm{vol}(\Phi_t(\mathcal{R})) = \int_{\mathcal{R}} |\det J_t(z)| dz = e^{-n\gamma t}\,\mathrm{vol}(\mathcal{R})$ where $\mathcal{R} \subset \mathbb{R}^{2n}$. This contraction is stronger as dimension increases. The conservative case is recovered with $\gamma = 0$ above; in this case, the symplectic structure is preserved and volumes remain invariant (Liouville's theorem). A known and interesting property of conformal Hamiltonian systems is that their Lyapunov exponents sum up in pairs to γ [31]. This imposes constraints on the admissible dynamics and controls the phase portrait near critical points. For other properties of attractor sets we refer to [32]. Finally, conformal symplectic transformations can be composed and form the so-called conformal group.

3 Conformal symplectic optimization

Consider (2.4) where we associate flows Φ_t^C and Φ_t^D to the respective vector fields $C(z)$ and $D(z)$. Conformal symplectic integrators can be constructed as *splitting methods* that approximate the true flow Φ_t by composing the individual flows Φ_t^C and Φ_t^D. Our procedure to obtain a numerical map Ψ_h, with step size $h > 0$, is to first obtain a numerical approximation to the conservative part of the system, $\dot{z} = \Omega \nabla H(z)$. This yields a numerical map Ψ_h^C that approximates Φ_h^C for small intervals of time $[t, t+h]$. One can choose any standard *symplectic integrator* for this task. Let us pick the simplest, i.e. the symplectic Euler method [30, pp. 189]. We thus have $\Psi_h^C : (x, p) \mapsto (X, P)$ where

$$X = x + h\nabla_p H(x, P), \qquad P = p - h\nabla_x H(x, P). \tag{3.1}$$

Now the dissipative part of the system, $\dot{z} = -\gamma D z$, can be integrated exactly. Indeed, $\dot{x} = 0$ and $\dot{p} = -\gamma p$, thus $\Psi_h^D : (x, p) = (x, e^{-\gamma h} p)$. With $\Psi_h \equiv \Psi_h^C \circ \Psi_h^D$ we obtain $\Psi_h : (x, p) \mapsto (X, P)$ as

$$P = e^{-\gamma h} p - h\nabla_x H(x, P), \qquad X = x + h\nabla_p H(x, P). \tag{3.2}$$

[4]It is not strictly necessary to be familiar with differential forms and exterior calculus to understand this paper. For the current purposes, it is enough to recall that the wedge product is a bilinear and antisymmetric operation, i.e. $dx \wedge (a dy + b dz) = a dx \wedge dy + b dx \wedge dz$ and $dx \wedge dy = -dy \wedge dx$ for scalars a and b and 1-forms dx, dy, dz (think about this as vector differentials); we refer to [29] and [30] for more details if necessary.

This is nothing but a *dissipative version* of the symplectic Euler method. Similarly, if we choose the leapfrog method [30, pp. 190] for Ψ_h^C and consider $\Psi_h \equiv \Psi_{h/2}^D \circ \Psi_h^C \circ \Psi_{h/2}^D$ we obtain

$$\tilde{X} = x + \frac{h}{2}\nabla_p H\big(\tilde{X}, e^{-\gamma h/2}p\big), \tag{3.3a}$$

$$\tilde{P} = e^{-\gamma h/2}p - \frac{h}{2}\big(\nabla_x H\big(\tilde{X}, e^{-\gamma h/2}p\big) + \nabla_x H\big(\tilde{X}, \tilde{P}\big)\big), \tag{3.3b}$$

$$X = \tilde{X} + \frac{h}{2}\nabla_p H(\tilde{X}, \tilde{P}), \tag{3.3c}$$

$$P = e^{-\gamma h/2}\tilde{P}. \tag{3.3d}$$

This is a *dissipative* version of the leapfrog, which is recovered when $\gamma = 0$. Note that in general (3.2) is implicit in P, and (3.3) is implicit in $\tilde{X}$ and P. However, both will become explicit for separable Hamiltonians, $H = T(p) + f(x)$, and in this case they are extremely efficient. Note also that (3.2) and (3.3) are completely general, i.e. by choosing a suitable Hamiltonian H one can obtain several possible optimization algorithms from these integrators. Next, we show important properties of these integrators. (Below we denote $t_k = kh$ for $k = 0, 1, \ldots$, $z_k \equiv z(t_k)$, etc.)

Definition 3.1 (Order of accuracy)**.** *A numerical map Ψ_h is said to be of order $r \geq 1$ if $\|\Psi_h(z) - \Phi_h(z)\| = O(h^{r+1})$ for any $z \in \mathbb{R}^{2n}$. (Recall that $h > 0$ is the step size and Φ_h is the true flow.)*

Definition 3.2 (Conformal symplectic integrator)**.** *A numerical map Ψ_h is said to be conformal symplectic if $z_{k+1} = \Psi_h(z_k)$ is conformal symplectic, i.e. $\omega_{k+1} = e^{-\gamma h}\omega_k$, whenever $\hat{\Phi}_h$ is applied to a smooth Hamiltonian. Iterating such a map yields $\omega_k = e^{-\gamma t_k}\omega_0$ so that* (2.6) *is preserved.*

Theorem 3.3. *Both methods* (3.2) *and* (3.3) *are conformal symplectic.*

Proof. Note that in both cases Ψ_h^C is a symplectic integrator, i.e. its Jacobian J_h^C obeys $(J_h^C)^T\Omega J_h^C = \Omega$—see (2.6) with $\gamma = 0$. Now the map Ψ_h^D defined above is conformal symplectic, i.e. one can verify that its Jacobian J_h^D obeys $(J_h^D)^T\Omega J_h^D = e^{-\gamma h}\Omega$. Hence, any composition of these maps will be conformal symplectic. For instance,

$$(J_h^C J_h^D)^T\Omega(J_h^C J_h^D) = (J_h^D)^T(J_h^C)^T\Omega J_h^C J_h^D = (J_h^D)^T\Omega J_h^D = e^{-\gamma h}\Omega. \tag{3.4}$$

The same would be true for any type of composition whose overall time step add up to h. □

Theorem 3.4. *The numerical scheme* (3.2) *is of order $r = 1$, while* (3.3) *is of order $r = 2$.*

Proof. The proof simply involves manipulating Taylor expansions for the numerical method and for the continuous-time system over a time interval of h; this is presented in Appendix A. □

We mention that one can construct higher order integrators by following the above approach, however these would be more expensive, involving more gradient computations per iteration. In practice, methods of order $r = 2$ tend to have the best cost benefit.

4 Symplectic structure of heavy ball and Nesterov

Consider the classical Hamiltonian (2.2) and replace into (3.2) to obtain

$$p_{k+1} = e^{-\gamma h} p_k - h\nabla f(x_k), \qquad x_{k+1} = x_k + \tfrac{h}{m} p_{k+1}, \tag{4.1}$$

where we now make the iteration number $k = 0, 1, \ldots$ explicit for convenience of the reader in relating to optimization methods. Introducing a change of variables,

$$v_k \equiv \frac{h}{m} p_k, \qquad \epsilon \equiv \frac{h^2}{m}, \qquad \mu \equiv e^{-\gamma h}, \tag{4.2}$$

we see that (4.1) is precisely the well-known CM method (1.1). Therefore, CM is nothing but a dissipative version of the symplectic Euler method. Thanks to Theorems 3.3 and 3.4 we have:

Corollary 4.1 (CM is "symplectic")**.** *The classical momentum or heavy ball method* (1.1) *is a conformal symplectic integrator for the Hamiltonian system* (2.2)*. Moreover, it is an integrator of order* $r = 1$*.*

Consider again the Hamiltonian (2.2) but replaced into (3.3). Let us also replace the last update (3.3d), i.e. from a previous iteration, into the first update (3.3a).[5] We thus obtain

$$x_{k+1/2} = x_k + \frac{h}{2m} e^{-\gamma h} p_k, \quad p_{k+1} = e^{-\gamma h} p_k - h\nabla f(x_{k+1/2}), \quad x_{k+1} = x_{k+1/2} + \frac{h}{2m} p_{k+1}. \tag{4.3}$$

Define

$$v_k \equiv \frac{h}{2m} p_k, \qquad \epsilon \equiv \frac{h^2}{2m}, \qquad \mu \equiv e^{-\gamma h}. \tag{4.4}$$

Then (4.3) can be written as

$$x_{k+1/2} = x_k + \mu v_k, \qquad v_{k+1} = \mu v_k - \epsilon\nabla f(x_{k+1/2}), \qquad x_{k+1} = x_{k+1/2} + v_{k+1}. \tag{4.5}$$

The reader can immediately recognize the close similarity with NAG (1.2); this would be exactly NAG if we replace $x_{k+1/2} \to x_k$ in the third update above. As we will show next, this small difference has actually profound consequences. Intuitively, by "rolling this last update backwards" one introduces a spurious friction into the method, as we will show through a symplectic perspective (Theorem 4.2 below). The method (4.3) is actually a second order accurate version of (4.1). In order to analyze the symplectic structure one must work on the phase space (x, p). The true phase space equivalent to NAG is given by

$$x_{k+1/2} = x_k + \frac{h}{m} e^{-\gamma h} p_k, \tag{4.6a}$$

$$p_{k+1} = e^{-\gamma h} p_k - h\nabla f(x_{k+1/2}), \tag{4.6b}$$

$$x_{k+1} = x_k + \frac{h}{m} p_{k+1}, \tag{4.6c}$$

which is completely equivalent to (1.2) under the correspondence (4.2).

[5] Note that it is valid to replace successive updates without changing the algorithm.

Theorem 4.2 (NAG is not "symplectic")**.** *Nesterov's accelerated gradient* (1.2)*, or equivalently* (4.6)*, is an integrator of order* $r = 1$ *to the Hamiltonian system* (2.2)*. This method is not conformal symplectic but rather contracts the symplectic form as*

$$\omega_{k+1} = e^{-\gamma h}\left[I - \frac{h^2}{m}\nabla^2 f(x_k)\right]\omega_k + O(h^3). \tag{4.7}$$

Proof. We work on phase space variables (x, p) thus NAG should be considered in the form (4.6). First we derive the order of accuracy of this method with respect to its underlying Hamiltonian system:

$$\dot{x} = \frac{p}{m}, \qquad \dot{p} = -\nabla f(x) - \gamma p. \tag{4.8}$$

Denote $x = x(t_k)$ and $p = p(t_k)$ and expand the exponential in (4.6a) to obtain $x_{k+1/2} = x + \frac{h}{m}p - \frac{h^2}{m}\gamma p + O(h^3)$. Using this and Taylor expansions in the last two updates of (4.6) yield

$$p_{k+1} = p - h\gamma p - h\nabla f(x) + \frac{h^2}{2}\gamma^2 p - \frac{h^2}{m}\nabla^2 f(x)p_k + O(h^3), \tag{4.9a}$$

$$x_{k+1} = x + \frac{h}{m}p - \frac{h^2}{m}\gamma p - \frac{h^2}{m}\nabla f(x) + O(h^3), \tag{4.9b}$$

where it is implicit that ∇f and $\nabla^2 f$ are computed at (x, p). From (4.8) we readily have

$$p(t_k + h) = p - h\nabla f - h\gamma p - \frac{h^2}{2m}\nabla^2 fp + \frac{h^2}{2}\gamma\nabla^2 fp + \frac{h^2}{2}\gamma\nabla f + \frac{h^2}{2}\gamma^2 p + O(h^3), \tag{4.10a}$$

$$x(t_k + h) = x + \frac{h}{m}p - \frac{h^2}{2m}\gamma p + O(h^3). \tag{4.10b}$$

Hence, by comparison with (4.9) we conclude that $x_{k+1} = x(t_k + h) + O(h^2)$ and $p_{k+1} = p(t_k + h) + O(h^2)$, which according to Definition 3.1 means that NAG is an integrator of order $r = 1$.

Second, we investigate how NAG deforms the symplectic structure. Consider the variational form of (4.6) (the notation is standard [30]):

$$dx_{k+1/2} = dx_k + \frac{h}{m}e^{-\gamma h}dp_k, \tag{4.11a}$$

$$dp_{k+1} = e^{-\gamma h}dp_k - h\nabla^2 f(x_{k+1/2})dx_{k+1/2}, \tag{4.11b}$$

$$dx_{k+1} = dx_k + \frac{h}{m}dp_{k+1}. \tag{4.11c}$$

Using these, bilinearity and the antisymmety of the wedge product, together the fact that $\nabla^2 f$ is symmetric, we obtain

$$\begin{aligned} dx_{k+1} \wedge dp_{k+1} &= dx_k \wedge dp_{k+1} \\ &= e^{-\gamma h}dx_k \wedge dp_k - hdx_k \wedge \nabla^2 f\big(x_{k+1/2}\big)dx_{k+1/2} \\ &= e^{-\gamma h}dx_k \wedge dp_k - \frac{h^2}{m}e^{-\gamma h}dx_k \wedge \nabla^2 f(x_{k+1/2})dp_k \\ &= e^{-\gamma h}dx_k \wedge dp_k - \frac{h^2}{m}e^{-\gamma h}dx_k \wedge \nabla^2 f(x_k)dp_k + O(h^3), \end{aligned} \tag{4.12}$$

where in the last passage we used a Taylor approximation for $x_{k+1/2}$. Thus, $dx_{k+1} \wedge dp_{k+1} \neq e^{-\gamma h} dx_k \wedge dp_k$, showing that the method is not conformal symplectic (see Definition 3.2). Moreover, using the symmetry of $\nabla^2 f$ we can write (4.12) as (4.7). □

While CM exactly preserve the same dissipation found in the underlying continuous-time system, NAG introduces some extra contraction or expansion of the symplectic form, depending whether $\nabla^2 f$ is positive definite or not. From (4.7), in k iterations of NAG, and neglecting the $O(h^3)$ error term, we have

$$\begin{aligned} \omega_k &\approx e^{-\gamma t_k} \prod_{i=1}^{k} \left[I - \frac{h^2}{m} \nabla^2 f(x_{k-i}) \right] \omega_0 \\ &\approx e^{-\gamma t_k} \left[I - \frac{h^2}{m} \big(\nabla^2 f(x_{k-1}) - \nabla^2 f(x_{k-2}) - \cdots - \nabla^2 f(x_0) \big) \right] \omega_0. \end{aligned} \tag{4.13}$$

This depends on the entire history of the Hessians from the initial point. Therefore, NAG contracts the symplectic form slightly more than the underlying conformal Hamiltonian system—assuming $\nabla^2 f$ is positive definite—and it does so in a way that depends on the Hessian of the objective function. Note that this is a small effect of $O(h^2)$. Moreover, if $\nabla^2 f$ has negative eigenvalues, e.g. f is nonconvex and has saddle points, then NAG actually introduces some spurious excitation in that direction. To gain some intuition, consider the simple case of a quadratic function:[6]

$$f(x) = (\lambda/2) x^2 \tag{4.14}$$

for some constant λ. Thus (4.7) becomes

$$\omega_{k+1} \approx e^{-\gamma h + \log(1 - h^2 \lambda/m)} \omega_k \approx e^{-(\gamma + h\lambda/m) h} \omega_k \quad \Longrightarrow \quad \omega_k \approx e^{-(\gamma + h\lambda/m) t_k} \omega_0. \tag{4.15}$$

This suggests that, effectively, the original damping of the system is being replaced by $\gamma \to \gamma + h\lambda/m$. Thus, if $\lambda > 0$ there is some spurious damping, whereas if $\lambda < 0$ there is some spurious excitation. We will confirm this conclusion from another perspective in Section 4.3 below.

4.1 Alternative form

It is perhaps more common to find Nesterov's method in the following form [2]:

$$x_{k+1} = y_k - \epsilon \nabla f(y_k), \qquad y_{k+1} = x_{k+1} + \mu_{k+1}(x_{k+1} - x_k), \tag{4.16}$$

where $\mu_{k+1} = k/(k+3)$. This is equivalent to (1.2), as can be seen by introducing the variable $v_k \equiv x_k - x_{k-1}$ and writing the updates in terms of x and v. When μ_k is constant, Theorem 4.2 shows that the method is not conformal symplectic. When $\mu_k = k/(k+3)$, the differential equation associated to (4.16) is equivalent to (2.1)/(2.2) with $\gamma = 3/t$. It is possible to generalize the above results for time dependent cases [18]. Therefore, also in this case, NAG does not preserve the symplectic structure; we note that (4.7) still holds with $e^{-\gamma h} \to e^{-3 \log(1 + h/t_k)}$ where $t_k = hk$.

[6] This quadratic function is actually enough to capture the behaviour when close to a critical point $x^\star$ since $f(x) \approx f(x^\star) + \frac{1}{2} \nabla^2 f(x^\star) x$ and one can work on rotated coordinates where $\nabla^2 f(x^\star) = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$.

4.2 Preserving stability and continuous-time rates

An important question is whether being "symplectic" is beneficial or not for optimization. Very recently, it has been shown [18] that symplectic discretizations of dissipative systems may indeed preserve continuous-time rates of convergence when f is smooth and the system is appropriately dampened (choice of γ); the continuous-time rates can be obtained via Lyapunov analysis. Thus, assuming that we have a suitable conformal Hamiltonian system, conformal symplectic integrators such as the general method (3.3), provide a principled approach to construct optimization algorithms that are guaranteed to respect the main properties of the system, such as stability of critical points and convergence rates. Furthermore, we claim that there is a delicate tradeoff where being conformal symplectic is related to an improved stability, in the sense that the method can operate with larger step sizes, while the spurious dissipation introduced by NAG (Theorem 4.2) may improve the convergence rate slightly, since it introduces more contraction, but at the cost of making the method less stable; we show these details in Section 6. Next, we also provide important additional insights into CM and NAG, such as their modified or perturbed equations and their *shadow Hamiltonians*, which describe these methods to a higher degree of resolution.

4.3 Shadow dynamical systems for Nesterov and heavy ball

We have shown above that both CM and NAG are a first order integrators to the conformal Hamiltonian system (4.8), however NAG changes slightly the behaviour of the original system since it introduces spurious damping or excitation. To understand its behaviour more closely, one can ask the following question: *for which continuous-time dynamical system NAG turns out to be a second order integrator?* In other words, we can look for a modified system that captures the behaviour of NAG more closely, up to $O(h^3)$. Every numerical method is known to have a modified or perturbed differential equation [30] (the brief discussion in [18] may also be useful). In answering this question, we thus find the following.

Theorem 4.3 (Shadow dynamical system for Nesterov's method)**.** *NAG* (1.2)*, or its equivalent phase space representation* (4.6)*, is a second order integrator to the following modified or perturbed equations:*

$$\dot{x} = \frac{1}{m}p - \frac{\gamma h}{2m}p - \frac{h}{2m}\nabla f(x), \qquad \dot{p} = -\nabla f(x) - \gamma p - \frac{h\gamma}{2}\nabla f - \frac{h}{2m}\nabla^2 f(x)p. \tag{4.17}$$

Proof. We look for vector fields $F(q,p;h)$ and $G(q,p;h)$ for the modified system

$$\dot{x} = \frac{p}{m} + hF, \qquad \dot{p} = -\nabla f(x) - \gamma p + hG, \tag{4.18}$$

such that (4.6) is an integrator of order $r = 2$. This can be done by computing [30]

$$F = \lim_{h\to 0} \frac{x_{k+1} - x(t_k + h)}{h^2}, \qquad G = \lim_{h\to 0} \frac{p_{k+1} - p(t_k + h)}{h^2}. \tag{4.19}$$

From (4.9) and (4.10) we obtain precisely (4.17). By the previously discussed approach through Taylor expansions one can also readily check that NAG is indeed an integrator of order $r = 2$ to this perturbed system. □

Note that we can combine (4.17) into a second order differential equation:

$$m\ddot{x} + m\left(\gamma I + \frac{h}{m}\nabla^2 f(x)\right)\dot{x} = -\left(I + \frac{h\gamma}{2}I - \frac{h^2\gamma^2}{4}I + \frac{h^2}{4m}\nabla^2 f(x)\right)\nabla f(x), \tag{4.20}$$

where I denotes the $n \times n$ identity matrix. We see that this equation has several new ingredients compared to

$$\ddot{x} + \gamma\dot{x} = -(1/m)\nabla f(x), \tag{4.21}$$

which is equivalent to (4.8). First, when $h \to 0$ the system (4.20) recovers (4.21), as it should since both must agree to leading order. Second, the spurious change in the damping coefficient reflects the behaviour of the symplectic form (4.7) (see also (4.15)). Third, we see that the gradient ∇f is rescaled by the contribution of several terms, including the Hessian $\nabla^2 f$, making explicit a curvature dependent behaviour, which also appears in the damping coefficient. Note that the modified equation (4.20), or equivalently (4.17), depends on the step size h, hence it captures an intrinsic behaviour of the discrete-time algorithm that is not captured by (4.8).

Since CM is also a first order integrator to (4.8), which is actually conformal symplectic, it is natural to consider its modified equation and compare with the one for NAG (4.17). We thus obtain the following.

Theorem 4.4 (Shadow Hamiltonian for heavy ball)**.** *The heavy ball or CM method* (1.1)*, equivalently written in the phase space as* (4.1)*, is a second order integrator to the following modified conformal Hamiltonian system:*

$$\dot{x} = \frac{1}{m}p - \frac{h\gamma}{2m}p - \frac{h}{2m}\nabla f(x), \qquad \dot{p} = -\nabla f(x) - \gamma p - \frac{h\gamma}{2}\nabla f(x) + \frac{h}{2m}\nabla^2 f(x)p. \tag{4.22}$$

Such a system admits the shadow Hamiltonian

$$\tilde{H} = \frac{1}{2m}\|p\|^2 + f(x) - \frac{h\gamma}{4m}\|p\|^2 - \frac{h}{2m}\langle\nabla f(x), p\rangle + \frac{h\gamma}{2}f. \tag{4.23}$$

Proof. It follows exactly as in Theorem 4.3. Also, one can readily verify that replacing (4.23) into (2.1) gives (4.23). □

We note the striking similarity between (4.22) and (4.17); the only difference is the sign of the last term in the second equation. Up to this level of resolution, the difference is that NAG introduces a spurious damping compared to CM, in agreement with the derivation of the symplectic form (4.7). On the other hand, notice that the perturbed system (4.22) for CM is conformal Hamiltonian, contrary to (4.17) that cannot be written in Hamiltonian form; this is the reason why structure-preserving discretizations tend to be more stable, since

the perturbed trajectories are always close, i.e. within a bounded error, from the original Hamiltonian dynamics. We can also combine (4.22) into

$$m\ddot{x} + m\gamma\dot{x} = -\left(I + \frac{h\gamma}{2}I - \frac{h^2\gamma^2}{4}I - \frac{h^2}{4m}\nabla^2 f(x)\right)\nabla f(x). \tag{4.24}$$

Again, this is strikingly similar to (4.20). Note that this equation does not have the spurious damping term $(h/m)\nabla^2 f(x)$ as in (4.20), making even more explicit that it preserves exactly the dissipation of the original continuous-time system. As we will show later, there is a balance between preserving such a dissipation and the stability of the method. While NAG introduces an extra damping, and may slightly help in an improved convergence since it dissipates more energy, this comes at the price in a decreased stability. Before showing this explicitly in Section 6, we first introduce a new optimization methods based on a relativistic system.

5 Dissipative relativistic optimization

Let us briefly mention some simple but fundamental concepts to motivate our approach. The previous algorithms are based on (2.2) which leads to a classical Newtonian system where time is just a parameter, independent of the Euclidean space where the trajectories live. This implies that there is no restriction on the speed, $\|v\| = \|dx/dt\|$, that a particle can attain. This translates to a discrete-time algorithm, such as (4.1), where large gradients ∇f give rise to a large momenta p, implying that the position updates for x can diverge. On the other hand, in special relativity, space and time form a unified geometric entity, the $(n+1)$-dimensional Minkowski spacetime with coordinates $X = (ct; x)$, where c denotes the speed of light. An infinitesimal distance on this manifold is given by $ds^2 = -(cdt)^2 + \|dx\|^2$. Null geodesics correspond to $ds^2 = 0$, implying $\|v\|^2 = \|dx/dt\|^2 = c^2$, i.e. no particle can travel faster than c. This imposes constraints on the geometry where trajectories take place—it is actually a hyperbolic geometry. With that being said, the idea is that by discretizing a relativistic system we can incorporate these features into an optimization algorithm which may bring benefits such as an improved stability.

A relativistic particle subject to a potential f is described by the following Hamiltonian:

$$H(x,p) = c\sqrt{\|p\|^2 + m^2c^2} + f(x). \tag{5.1}$$

In the classical limit, $\|p\| \ll mc$, one obtains $H = mc^2 + \|p\|^2/(2m) + f(x) + O(1/c^2)$, recovering (2.2) up to the constant $E_0 = mc^2$, which has no effect in deriving the equations of motion. Replacing (5.1) into (2.1) we thus obtain a *dissipative relativistic system*:

$$\dot{x} = \frac{cp}{\sqrt{\|p\|^2 + m^2c^2}}, \qquad \dot{p} = -\nabla f - \gamma p. \tag{5.2}$$

Importantly, in (5.2) the momentum is normalized by the $\sqrt{\cdot}$ factor, so $\dot{x}$ remains bounded even if p was to go unbounded. Now, replacing (5.1) into the first order accurate conformal

symplectic integrator (3.2), we readily obtain

$$p_{k+1} = e^{-\gamma h} p_k - h\nabla f(q_k), \qquad x_{k+1} = x_k + \frac{hcp_{k+1}}{\sqrt{\|p_{k+1}\|^2 + m^2c^2}}. \tag{5.3}$$

When $c \to \infty$ the above updates recover CM (4.1). Thus, this method is a relativistic generalization of CM or heavy ball. Moreover, the method (5.3) is a first order conformal symplectic integrator by construction (see Theorems 3.3 and 3.4).

One can replace the Hamiltonian (5.1) into (3.3) to obtain a second order version of (5.3). However, motivated by the close connection between NAG and (4.3)—recall the comments following (4.5) about NAG "rolling back" the last update—let us additionally introduce a convex combination, $\alpha x_{k+1/2} + (1-\alpha)x_k$ where $0 \leq \alpha \leq 1$, between the initial and midpoint of the method. In this manner, we can interpolate between a conformal symplectic regime and a spurious Hessian damping regime (recall Theorem 4.2). Therefore, we obtain the following integrator:

$$x_{k+1/2} = x_k + (hc/2)e^{-\gamma h/2} p_k \Big/ \sqrt{e^{-\gamma h}\|p_k\|^2 + m^2c^2}, \tag{5.4a}$$

$$p_{k+1/2} = e^{-\gamma h/2} p_k - h\nabla f(x_{k+1/2}), \tag{5.4b}$$

$$x_{k+1} = \alpha x_{k+1/2} + (1-\alpha)x_k + (hc/2)p_{k+1/2} \Big/ \sqrt{\|p_{k+1/2}\|^2 + m^2c^2}, \tag{5.4c}$$

$$p_{k+1} = e^{-\gamma h/2} p_{k+1/2}. \tag{5.4d}$$

We call this method *Relativistic Gradient Descent* (RGD). By introducing

$$v_k \equiv \frac{h}{2m} p_k, \qquad \epsilon \equiv \frac{h^2}{2m}, \qquad \mu \equiv e^{-\gamma h}, \qquad \delta \equiv \frac{4}{c^2 h^2}, \tag{5.5}$$

the updates (5.4) assume the equivalent form stated in Algorithm 1 in the introduction.

RGD (5.4) (resp. Algorithm 1) has several interesting limits, recovering the behaviour of known algorithms as particular cases. For instance, when $c \to \infty$ (resp. $\delta \to 0$) it reduces to an interpolation between CM (4.1) (resp. (1.1)) and NAG (4.6) (resp. (1.2)). If we additionally set $\alpha = 0$ it becomes precisely NAG, whether when $\alpha = 1$ it becomes a second order version (in terms of accuracy) of CM.[7] When $\alpha = 1$, and arbitrary c (or δ), RGD is a conformal symplectic integrator thanks to Theorems 3.3. Recall also that Theorem 3.4 implies that RGD is a second order accurate integrator. When $\alpha = 0$, and arbitrary c (or δ), RGD is no longer conformal symplectic and introduces a Hessian driven damping in the spirit of NAG. Finally, the parameter c (or δ) controls the strength of the normalization term in the position updates of (5.4) (or Algorithm 1), which can help preventing divergences when navigating through a rough landscape with large gradients, or fast growing tails. Indeed, note that $\|x_{k+1} - \alpha x_{k+1/2} - (1-\alpha)x_k\| \leq 1/\delta$ is always bounded for $\delta > 0$; this becomes unbounded when $\delta \to 0$, i.e. in the classical limit of CM and NAG.

In short, RGD is a novel algorithm with quite some flexibility and unique features, generalizing perhaps the two most important accelerated gradient based methods in the

[7] The dynamics of both CM and this second order version is pretty close, and if anything the latter is even more stable than the former (see Section 6).

literature, which can be recovered as limiting cases. Next, we illustrate numerically through simple yet insightful examples that RGD can be more stable and faster than CM and NAG.

6 Tradeoff between stability and convergence rate

Here we illustrate an interesting phenomenon: there is a tradeoff between stability versus convergence rate. Intuitively, an improved rate is associated to a higher "contraction," i.e. the introduction of spurious dissipation in the numerical method. However, this makes the method less stable, and ultimately very sensitive to parameter tuning. On the other hand, a geometric or structure-preserving integrator may have slightly less contraction, since it preserves the original dissipation of the continuous-time system exactly, but it is more stable and able to operate with larger step sizes. Furthermore, a structure-preserving method is guaranteed to reproduce very closely, perhaps even up to a negligible error, the continuous-time rates of convergence [18]. This indicates that there may have benefits in considering this class of methods for optimization, such as conformal symplectic integrators that are being advocated in this paper.

Stability of a numerical integrator means the region of hyperparameters, e.g. values of the step size, such that the method is able to converge. The larger this region, more stable is the method. The convergence rate is a measure of how fast the method tends to the minimum, and this is related to the amount of contraction between subsequent states, or subsequent values of the objective function. For instance, since NAG introduces some spurious dissipation—recall (4.7) and (4.20)—we expect that it may have a slightly higher contraction compared to CM, which exactly preserves the dissipation of the continuous-time system—recall (4.24). Thus, such a spurious dissipation can induce a slightly improved convergence rate, but as we will show below, at the cost of making the method more unstable and thus requiring smaller step sizes.

Let us consider a standard linear stability analysis, which involves a quadratic function (4.14) such that the previous methods can be treated analytically. Thus, replacing (4.14) into CM in the form (4.1) it is possible to write the algorithm as a linear system:

$$z_{k+1} = T_{\mathrm{CM}} z_k, \qquad T_{\mathrm{CM}} = \begin{bmatrix} 1 - h^2\lambda/m & (h/m)e^{-\gamma h} \\ -h\lambda & e^{-\gamma h} \end{bmatrix}, \tag{6.1}$$

where we denote $z = \left[\begin{smallmatrix} x \\ p \end{smallmatrix}\right]$. Similarly, NAG in the form (4.6) yields

$$z_{k+1} = T_{\mathrm{NAG}} z_k, \qquad T_{\mathrm{NAG}} = \begin{bmatrix} 1 - h^2\lambda/m & (h/m)e^{-\gamma h}(1 - h^2\lambda/m) \\ -h\lambda & e^{-\gamma h}(1 - h^2\lambda/m) \end{bmatrix}, \tag{6.2}$$

while RGD (5.4), with $c \to \infty$ and $\alpha = 1$, yields[8]

$$z_{k+1} = T_{\mathrm{RGD}} z_k, \qquad T_{\mathrm{RGD}} = \begin{bmatrix} 1 - h^2\lambda/(2m) & h/(2m)e^{-\gamma h/2}(2 - h^2\lambda/(2m)) \\ -h\lambda e^{-\gamma h/2} & e^{-\gamma h}(1 - h^2\lambda/(2m)) \end{bmatrix}. \tag{6.3}$$

[8]The case of finite c is nonlinear and not amenable to such an analysis. However, the case $c \to \infty$ already provides useful insights.

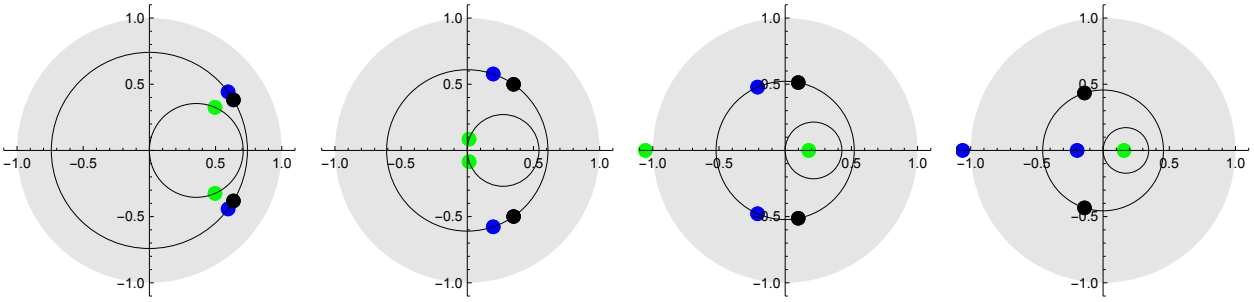


Figure 1: Stability of CM (4.1) (*blue*), NAG (4.6) (*green*), and RGD (5.4) with $c \to \infty$ and $\alpha = 1$ (*black*)—in this case it becomes a dissipative version of the Leapfrog to system (4.8). We plot the eigenvalues in the complex plane; x-axis is the real part, y-axis is the imaginary part. The unit circle represent the stability region, i.e. once an eigenvalue leaves the gray area the corresponding method becomes unstable. Both CM and RGD are symplectic thus their eigenvalues always move on a circle of radius $e^{-\gamma h/2}$ centered at the origin. NAG has eigenvalues in the smaller circle with radius $1/(e^{\gamma h}+1)$ and centered at $1/(e^{\gamma h}+1)$ on the x-axis; the circle is dislocated from the origin precisely due to spurious dissipation. From left to right we increase the step size h while keeping γ, m, and λ fixed. As h increases the eigenvalues move on the circles in the counterclockwise direction until they fall on the real line. Eventually they leave the unit circle and the associated method becomes unstable. Note how CM has higher stability than NAG, and RGD has even higher stability than CM.

A linear system is stable if the spectral radius of its transition matrix is $\rho(T) \leq 1$. We can compute the eigenvalues of the above matrices and check for which range of parameters they remain inside the unit circle; e.g. for given γ, m, and λ we can find the allowed range of the step size h for which the maximum eigenvalue in absolute value is $|\lambda_{\max}| \leq 1$. Instead of showing the explicit formulas for these eigenvalues, which can be obtained quite simply but are cumbersome, let us illustrate what happens graphically.

In Fig. 1, the shaded gray area represents the unit circle. Any eigenvalue that leaves this area makes the associated algorithm unstable. Here we fix $m = \lambda = \gamma = 1$ (other choices are equivalent) and we vary the step size $h > 0$. These eigenvalues are in general complex and lie on a circle which is determined by the amount of friction in the system. Note how for CM and RGD this circle is centered at the origin, with radius $\sqrt{\mu} \equiv e^{-\gamma h/2}$, since these methods are conformal symplectic and exactly preserve the dissipation of the underlying continuous-time system. However, NAG introduces a spurious damping which is reflected as the circle being translated from the center, at a distance $1/(e^{\gamma h}+1)$, and moreover this circle has a smaller radius of $1/(e^{\gamma h}+1)$ compared to CM and RGD; since this radius is smaller, NAG may have a faster convergence when these eigenvalues are complex. As we increase h (left to right in Fig. 1), the eigenvalues move counterclockwise on the circles until falling on the real line, where one of them goes to the left while the other goes to the right. Eventually, the leftmost eigenvalue leaves the unit circle for a large enough h (third panel in Fig. 1). Note that NAG becomes unstable first, followed by CM, and only then by RGD. The main point is that CM and RGD can still be stable for much larger step sizes compared

to NAG, and RGD is even more stable than CM as seen in the rightmost plot in Fig. 1; this is a consequence of RGD being an integrator of order $r = 2$ whereas CM is of order $r = 1$. Hence, even though NAG may have a slightly faster convergence (due to a stronger contraction), it requires a smaller step sizes and its stability is more sensitive compared to a conformal symplectic method. On the other hand, both CM and RGD can operate with larger step sizes, which in practice may even result in a faster solver compared to NAG.

To provide a more quantitative statement, after computing the eigenvalues of the above transition matrices for given $\mu \equiv e^{-\gamma h}$, m, and λ, we find the following threshold for stability:

$$h_{\mathrm{CM}} \leq \sqrt{m(1+\mu+\mu^2+\mu^3)}\big/(\mu\sqrt{\lambda}), \tag{6.4}$$

$$h_{\mathrm{NAG}} \leq \sqrt{m(1+\mu+\mu^2+\mu^3)}\big/\sqrt{\mu\lambda(1+\mu+\mu^2)}, \tag{6.5}$$

$$h_{\mathrm{RGD}} \leq \sqrt{2m(1+\mu+\mu^2+\mu^3)}\big/\sqrt{\mu\lambda(1+\mu)}. \tag{6.6}$$

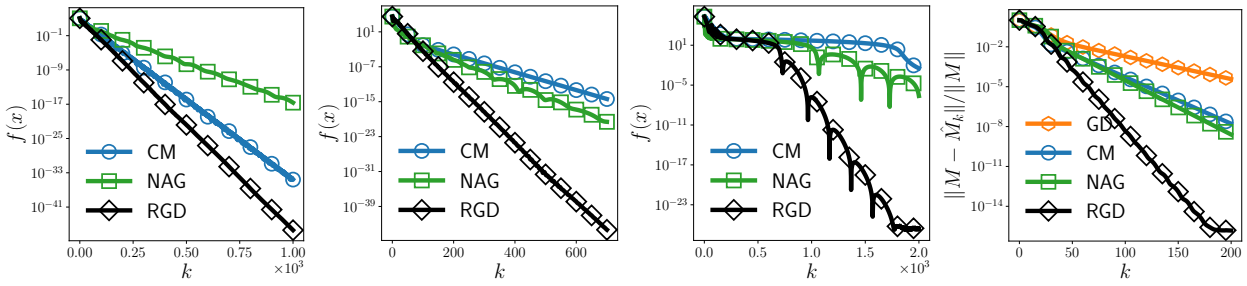
We can clearly see that RGD has the largest region for h, followed by CM, then by NAG, in agreement with the results of Fig. 1.

7 Numerical experiments

Let us compare RGD (Algorithm 1) against NAG (1.2) and CM (1.1) on some test problems. We stress that all hyperparameters of each of these methods were systematically optimized through Bayesian optimization [33] (the default implementation uses a Tree of Parzen estimators). This yields *optimal* and *unbiased* parameters automatically. Moreover, by checking the distribution of these hyperparameters during the tuning process we can get intuition on the sensitivity of each method. Thus, for each algorithm, we show its convergence rate in Fig. 2 when the best hyperparameters were used. In addition, in Fig. 3 we show the distribution of hyperparameters during the Bayesian optimization step—the parameters are indicated and color lines follow Fig. 2. Such values are obtained only when the respective algorithm was able to converge. We note that usually CM and NAG diverged more often than RGD which seemed more robust to parameter choice. Below we describe some of the optimization problems where such algorithms were tested over. In Appendix B we provide several additional experiments illustrating the benefits of RGD. The actual code related to our implementation is extremely simple and can be found at [34].

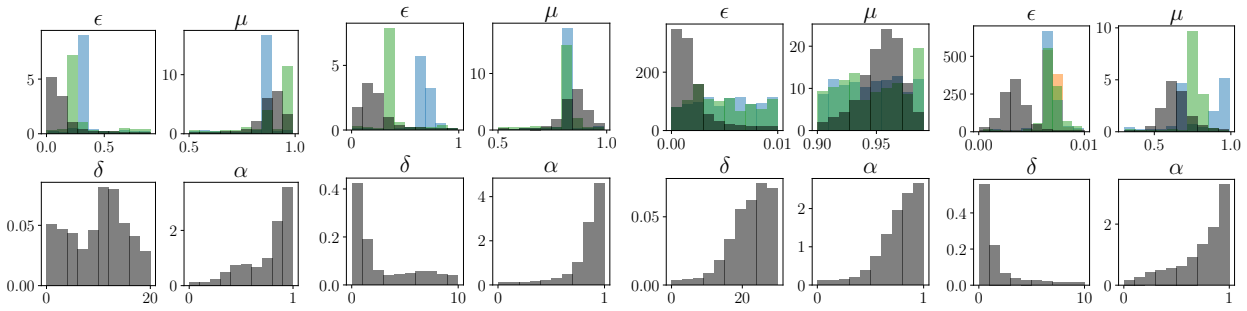
7.1 Correlated quadratic

Consider $f(x) = (1/2)x^T Q x$ where $Q_{ij} = \rho^{|i-j|}$, $\rho = 0.95$, and Q has size 50×50—this function was also used in [14]. We initialize the position at random, $x_{0,i} \sim \mathcal{N}(0, 10)$, and the velocity as $v_0 = 0$. The convergence results are shown in Fig. 2a. The distribution of parameters during tuning are in Fig. 3a, showing that $\alpha \to 1$ is preferable. This gives evidence for an advantage in being conformal symplectic. Note also that $\delta > 0$, thus "relativistic effects" played a role in improving convergence.



(a) Correlated quadratic (b) Random quadratic (c) Rosenbrock (d) Matrix completion

Figure 2: Convergence rate showing improved performance of RGD (Algorithm 1); see text.



(a) Correlated quadratic (b) Random quadratic (c) Rosenbrock (d) Matrix completion

Figure 3: Histograms of hyperparameter tuning by Bayesian optimization. Tendency towards $\alpha \approx 1$ indicates benefits of being symplectic, while $\alpha \approx 0$ of being extra damped as in NAG. Tendency towards $\delta > 0$ indicates benefits of relativistic normalization. (Colors follow Fig. 2.)

7.2 Random quadratic

Consider $f(q) = (1/2)x^T Q x$ where Q is a 500×500 positive definite random matrix with eigenvalues uniformly distributed in $[10^{-3}, 10]$. Convergence rates are in Fig. 2b with the histograms of parameter search in Fig. 3b. Again, there is a preference towards $\alpha \to 1$, evidencing benefits in being conformal symplectic.

7.3 Rosenbrock

For a challenging problem in higher dimensions, consider the nonconvex Rosenbrock function $f(x) \equiv \sum_{i=1}^{n-1} \left(100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right)$ with $n = 100$ [35,36]; this case was already studied in detail [37]. Its landscape is quite involved, e.g. there are two minimizers, one global at $x^\star = (1, \ldots, 1)^T$ with $f(x^\star) = 0$ and one local near $x \approx (-1, 1, \ldots, 1)^T$ with $f \approx 3.99$. There are also—exponentially—many saddle points [37], however only two of these are actually hard to escape. These four stationary points account for 99.9% of the solutions found by Newton's method [37]. We note that both minimizers lie on a flat, deep, and narrow valley, making optimization challenging. In Fig. 2c we have the convergence of each method

initialized at $x_{0,i} = \pm 2$ for i odd/even. Fig. 3c shows histograms for parameter selection. Again, we see the favorable symplectic tendency, $\alpha \to 1$. Here relativistic effects, $\delta \neq 0$, played a predominant role in the improved convergence of RGD.

7.4 Matrix completion

Consider an $n \times n$ matrix M of rank $r \ll n$ with observed entries in the support $(i, j) \in \Omega$, where $P_\Omega(M)_{ij} = M_{ij}$ if $(i, j) \in \Omega$ and $P_\Omega(M)_{ij} = 0$ projects onto this support. The goal is to recover M from the knowledge of $P_\Omega(M)$. We assume that the rank r is known. In this case, if the number of observed entries is $O(rn)$ it is possible to recover M with high probability [38]. We do this by solving the nonconvex problem $\min_{U,V} \|P_\Omega(M - UV^T)\|_F^2$, where $U, V \in \mathbb{R}^{n\times r}$, by *alternating minimization*: for each iteration we apply the previous algorithms first on U with V held fixed, followed by similar updates for V with the new U fixed. This is a know technique for gradient descent (GD), which we additionally include as a baseline. We generate $M = RS^T$ where $R, S \in \mathbb{R}n \times r$ have iid entries from the normal distribution $\mathcal{N}(1, 2)$. We initialize U and V sampled from the standard normal. The support is chosen uniformly at random with sampling ratio $s = 0.3$, yielding $p = sn^2$ observed entries. We set $n = 100$ and $r = 5$. This gives a number of effective degrees of freedom $d = r(2n - r)$ and the "hardness" of the problem can be quantified via $d/p \approx 0.325$. Fig. 2d shows the convergence rate, and Fig. 3d the parameter search.

8 Discussion and outlook

This paper introduces a new perspective on a recent line of research connecting accelerated optimization methods to continuous-time dynamical systems that have been playing a major role in machine learning. We brought *conformal symplectic* techniques for *dissipative systems* into this context, besides proposing a new method called *Relativistic Gradient Descent* (RGD), based on a dissipative relativistic system; see Algorithm 1. RGD generalizes both the classical momentum (CM) or heavy ball method—given by (1.1)—as well as Nesterov's accelerated gradient (NAG)—given by (1.2); each of these methods are recovered as particular cases from RGD which has no additional computational cost compared to CM and NAG. Moreover, RGD has more flexibility, can interpolate between a conformal symplectic behaviour or introduce some Hessian dependent damping in the spirit of NAG, and has potential to control instabilities due to large gradients by normalizing the momentum. In our experiments, RGD significantly outperformed CM and NAG, specially in settings with large gradients or functions with a fast growth; besides Section 7 we report several additional examples in Appendix B.

We also elucidated what is the symplectic structure behind CM and NAG. We found that the former turns out to be a conformal symplectic integrator (Corollary 4.1), thus being "dissipative-preserving," while the latter introduces a spurious contraction of the symplectic form by a Hessian driven damping (Theorem 4.2). This is an effect of second order in the step size but may affect convergence and stability. We pointed out a tradeoff between this extra

contraction and the stability of a conformal symplectic method. We also derived modified or perturbed equations for CM and NAG, describing these methods to a higher degree of resolution; this analysis provides several new insights into these methods and may form the basis for exploring these algorithms using different techniques compared to standard approaches in pure optimization.

On a higher level, this paper shows how structure-preserving discretizations of classical dissipative systems can be useful for studying existing optimization algorithms, as well as introduce new methods inspired by real physical systems. A thorough justification for the use of structure-preserving—or "dissipative symplectic"—discretizations in this context was recently provided in [18] under great generality.

Finally, a more refined analysis of RGD is certainly an interesting future problem, though considerably challenging due to the nonlinearity introduced by the $\sqrt{1+\delta\|v\|^2}$ term in the updates of Algorithm 1. To give an example, even if one assumes a simple quadratic function $f(x) = (\lambda/2)x^2$, the differential equation (5.2) is nonlinear and does not admit a closed form solution, contrary to the differential equation associated to CM and NAG which is linear and can be readily integrated. Thus, even in continuous-time, the analysis for RGD is likely to be involved. Finally, it would be interesting to consider RGD in a stochastic setting, namely investigate its diffusive properties in a random media, which may bring benefits to nonconvex optimization and sampling.

Acknowledgments

GF would like to thank Michael I. Jordan for discussions. This work was supported by grants ARO MURI W911NF-17-1-0304, NSF 2031985, and NSF 1934931.

A Order of accuracy of the general integrators

It is known that a composition of the type $\Psi_h^A \circ \Psi_h^B$, where A and B represents the components of distinct vector fields, leads to an integrator of order $r = 1$, whereas a composition in the form $\Psi_{h/2}^A \circ \Psi_h^B \circ \Psi_{h/2}^A$ leads to an integrator of order $r = 2$ [30]—the latter is known as Strang splitting. However, here we provide an explicit and direct proof of these facts for the generic integrators (3.2) and (3.3), respectively.

Proof of Theorem 3.4. From the equations of motion (2.1) and Taylor expansions:

$$\begin{aligned} x(t_k+h) &= x + h\dot{x} + \frac{h^2}{2}\ddot{x} + O(h^3) \\ &= x + h\nabla_p H + \frac{h^2}{2}\left(\nabla_{xp}^2 H\dot{x} + \nabla_{pp}^2 H\dot{p}\right) + O(h^3) \\ &= x + h\nabla_p H + \frac{h^2}{2}\nabla_{xp}^2 H\nabla_p H - \frac{h^2}{2}\nabla_{xp}^2\nabla_x H - \frac{h^2}{2}\gamma\nabla_{pp}^2 Hp + O(h^3), \end{aligned} \tag{A.1}$$

and

$$\begin{aligned} p(t_k+h) &= p + h\dot{p} + \frac{h^2}{2}\ddot{p} + O(h^3) \\ &= p - h\nabla_x H - h\gamma p + \frac{h^2}{2}\left(-\nabla^2_{xx}H\dot{x} - \nabla^2_{xp}H\dot{p} - \gamma\dot{p}\right) + O(h^3) \\ &= p - h\nabla_x H - h\gamma p - \frac{h^2}{2}\nabla^2_{xx}H\nabla_p H + \frac{h^2}{2}\nabla^2_{xp}H\nabla_x H + \frac{h^2}{2}\gamma\nabla^2_{xx}Hp \\ &\quad + \frac{h^2}{2}\gamma\nabla_x H + \frac{h^2}{2}\gamma^2 p + O(h^3), \end{aligned} \tag{A.2}$$

where we denote $x \equiv x(t_k)$ and $p \equiv p(t_k)$ for $t_k = kh$ $(k = 0, 1, \dots)$, and it is implicit that all gradients and Hessians of H are being computed at (x, p).

Consider (3.2). Under one step of this map, starting from the point (x, p), upon using Taylor expansions we have

$$x_{k+1} = x + h\nabla_p H + O(h^2) \tag{A.3}$$

and

$$p_{k+1} = e^{-\gamma h}p - h\nabla_x H + O(h^2) = p - \gamma h p - h\nabla_x H(x,p) + O(h^2). \tag{A.4}$$

Comparing these last two equations with (A.1) and (A.2) we conclude that

$$x_{k+1} = x(t_k + h) + O(h^2), \qquad p_{k+1} = p(t_k + h) + O(h^2). \tag{A.5}$$

Therefore, the discrete state approximates the continuum state up to an error of $O(h^2)$, obeying Definition 3.1 with $r = 1$.

The same approach is applicable to the numerical map (3.3). Expanding the first update:

$$\begin{aligned} \tilde{X} &= x + \frac{h}{2}\nabla_p H\big(x + (h/2)\nabla_p H, p - (h/2)\gamma p\big) + O(h^3), \\ &= x + \frac{h}{2}\nabla_p H + \frac{h^2}{4}\nabla^2_{xp}H\nabla_p H - \frac{h^2}{4}\gamma\nabla^2_{pp}Hp + O(h^3). \end{aligned} \tag{A.6}$$

Expanding the second update:

$$\begin{aligned} \tilde{P} &= e^{-\gamma h/2}p - \frac{h}{2}\nabla_x H\big(x + (h/2)\nabla_p H, p - (h/2)\gamma p\big) \\ &\quad - \frac{h}{2}\nabla_x H\big(x + (h/2)\nabla_p H, p - (h/2)\gamma p - h\nabla_x H\big) + O(h^3), \\ &= e^{-\gamma h/2}p - h\nabla_x H - \frac{h^2}{2}\nabla^2_{xx}H\nabla_p H + \frac{h^2}{2}\gamma\nabla^2_{xp}Hp + \frac{h^2}{2}\nabla^2_{xp}H\nabla_x H + O(h^3). \end{aligned} \tag{A.7}$$

Making use of (A.6) and (A.7) we thus find:

$$\begin{aligned} X &= \tilde{X} + \frac{h}{2}\nabla_p H(\tilde{X}, \tilde{P}) \\ &= x + \frac{h}{2}\nabla_p H + \frac{h^2}{4}\nabla^2_{xp}H\nabla_p H - \frac{h^2}{4}\gamma\nabla^2_{pp}Hp \\ &\quad + \frac{h}{2}\nabla H\big(x + (h/2)\nabla_p H, p - (h/2)\gamma p - h\nabla_x H\big) + O(h^3) \\ &= x + h\nabla_p H + \frac{h^2}{2}\nabla^2_{xp}H\nabla_p H - \frac{h^2}{2}\gamma\nabla^2_{pp}Hp - \frac{h^2}{2}\nabla^2_{pp}H\nabla_x H + O(h^3). \end{aligned} \tag{A.8}$$

Comparing with (A.1) we conclude that

$$x_{k+1} = x(t_k + h) + O(h^3). \tag{A.9}$$

Finally, from (A.7) we have

$$\begin{aligned} P &= e^{-\gamma h/2}\tilde{P} \\ &= e^{\gamma h}p - e^{-\gamma h/2}\Big\{h\nabla_x H + \frac{h^2}{2}\nabla^2_{xx}H\nabla_p H + \frac{h^2}{2}\gamma\nabla^2_{xp}Hp - \frac{h^2}{2}\nabla^2_{xp}H\nabla_x H\Big\} + O(h^3) \\ &= p - \gamma hp + \frac{h^2}{2}\gamma^2 p - h\nabla_x H - \frac{h^2}{2}\nabla^2_{xx}H\nabla_p H + \frac{h^2}{2}\gamma\nabla^2_{xp}Hp \\ &\quad + \frac{h^2}{2}\nabla^2_{xp}H\nabla_x H + \frac{h^2}{2}\gamma\nabla_x H + O(h^3). \end{aligned} \tag{A.10}$$

Comparing this with (A.2) implies

$$p_{k+1} = p(t_k + h) + O(h^3). \tag{A.11}$$

Therefore, in this case we satisfy Definition 3.1 with $r = 2$. □

From the above general results it is immediate that:

- CM (1.1)—or equivalently (4.1) which is more appropriate to make connections with the continuous-time system—is a first order integrator to the conformal Hamiltonian system (2.1) with the classical Hamiltonian (2.2); the equations of motion are explicitly given by (4.8).
- The relativistic extension of CM given by (5.3) is a first order integrator to the conformal relativistic Hamiltonian system (5.2).
- RGD (5.4) with $\alpha = 1$—also equivalently written as Algorithm. 1—is a second order integrator to system (5.2).

B Additional numerical experiments

Here we compare RGD (Algorithm 1) with CM (1.1) and NAG (1.2) on several additional test functions; for details on these functions see e.g. [39] and references therein. We follow the procedure already described in Section 7 where we optimized the hyperparameters of these algorithm using Bayesian optimization.[9] We report the convergence rate using the best parameters found together with histograms of the parameter search. In all cases we initialize the velocity as $v_0 = 0$. The initial position x_0 was chosen inside the range where the corresponding test function is usually considered.

[9]We provide the actual code used in our numerical simulations in [34].

First we consider functions with a quadratic growth. These results are shown in Figs. 4–7. In this case RGD performed similarly to CM and NAG, although with some improvement. In any case RGD proved to be more stable, i.e. it worked well for a wider range of hyperparameters.

We expect that RGD stands out on settings with large gradients or objective functions with fast growing tails. Therefore, in the remaining figures, i.e. Fig. 8–15, we consider more challenging optimization problems with functions that grow stronger than a quadratic. For some of these problems the minimum lies on a flat valley, making it hard for an algorithm to stop around the minimum after gaining a lot of speed from a very steep descent direction. Note that in all these cases the improvement of RGD over CM and NAG is significant, and the parameter δ—which controls relativistic effects—had an important role. The conformal symplecticity, which is indicated by the tendency $\alpha \to 1$, also brings an improved stability in the discretization. These results provide compelling evidence for the benefits of RGD.

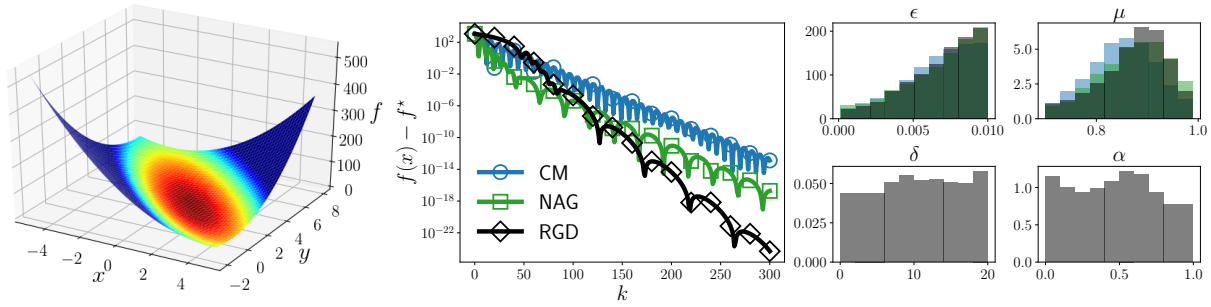


Figure 4: *Booth function*: $f(x, y) \equiv (x + 2y - 7)^2 + (2x + y - 5)^2$. Global minimum at $f(1, 3) = 0$. We initialize at $x_0 = (10, 10)$. This function is usually evaluated on the region $-10 \leq x, y \leq 10$. All methods perform well on this problem which is not challenging.

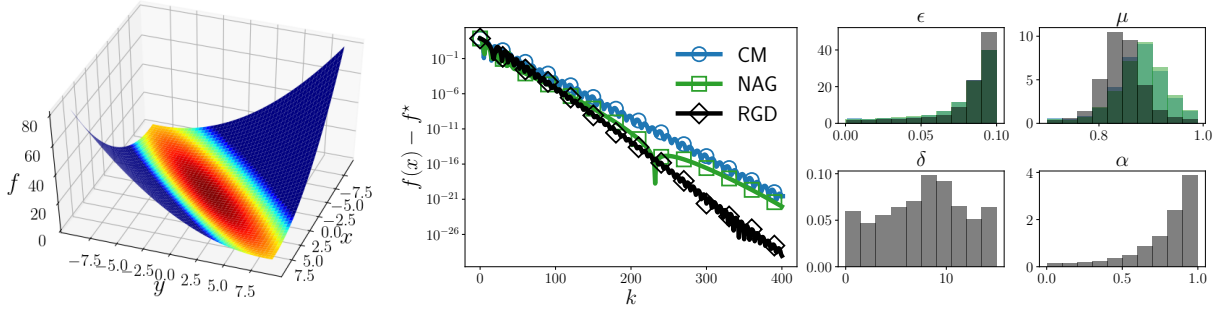


Figure 5: *Matyas function*: $f(x, y) \equiv 0.26(x^2 + y^2) - 0.48xy$. Global minimum is at $f(0, 0) = 0$. We initialize at $x_0 = (10, -7)$. This function is usually evaluated on the region $-10 \leq x, y \leq 10$. Even though the function has a—not so strong—quadratic growth, we see a slight improvement of RGD; note $\delta > 0$. Note also the "symplectic tendency" $\alpha \to 1$.

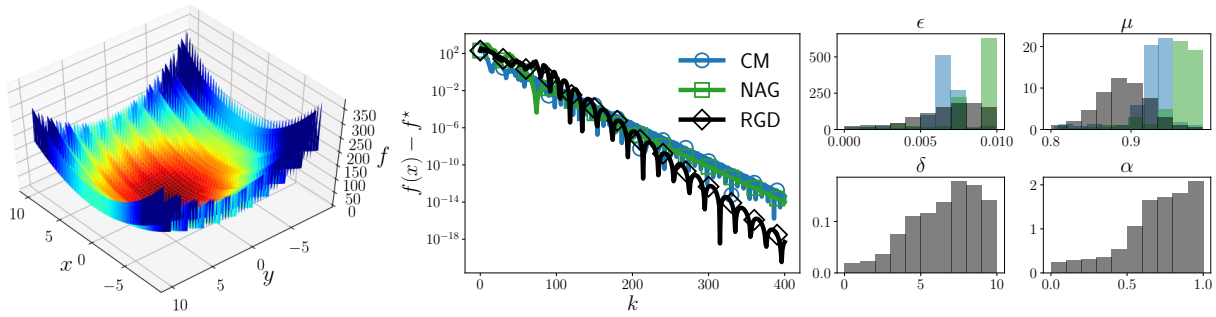


Figure 6: *Lévi function #13*: $f(x,y) \equiv \sin^2 3\pi x + (x-1)^2(1+\sin^2 3\pi y) + (y-1)^2(1+\sin^2 2\pi y)$. It is multimodal with the global minimum at $f(1,1) = 0$. We initialize at $x_0 = (10,-10)$. This function is usually studied on the region $-10 \leq x, y \leq 10$. Although this function is nonconvex, the optimization problem is not very challenging. However, we noticed that CM and NAG got stuck on a local minimum more often than RGD when running this example multiple times.

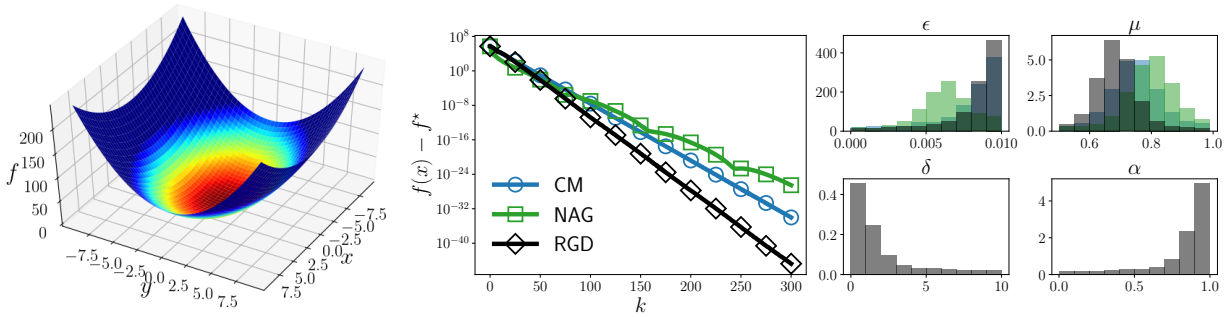


Figure 7: *Sum of squares*: $f(x) \equiv \sum_{i=1}^{n} i x_i^2$. The minimum is at $f(0) = 0$. We consider $n = 100$ dimensions and initialize at $x_0 = (10, \ldots, 10)$. The usual region of study is $-10 \leq x_i \leq 10$. Note that there is a clear tendency towards $\alpha \to 1$ in this case, i.e. in being conformal symplectic.

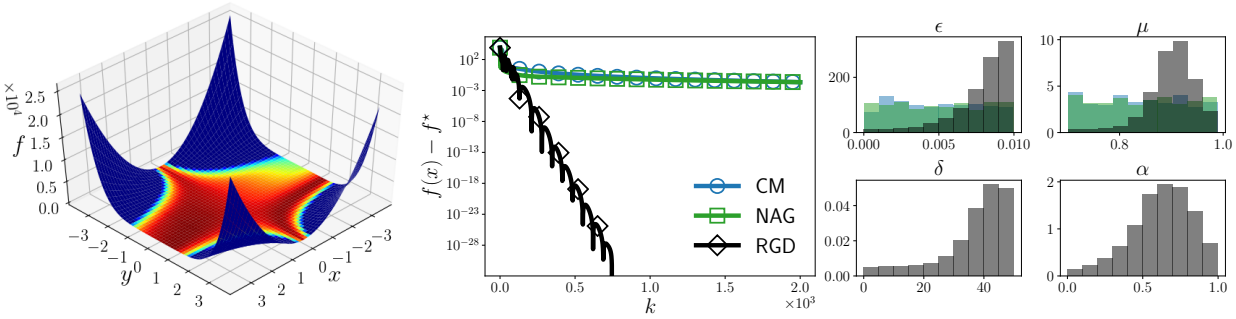


Figure 8: *Beale function*: $f(x,y) \equiv (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2$. The global minimum is at $f(3, 1/2) = 0$, lying on a flat and narrow valley which makes optimization challenging. Note also that this functions grows stronger than a quadratic. This function is usually considered on the region $-4.5 \leq x, y \leq 4.5$. We initialize at $x_0 = (-3,-3)$. Note how CM and NAG were unable to minimize the function, while RGD was able to find the global minimum to high accuracy; $\delta \gg 0$ played a predominant role, indicating benefits from "relativistic effects."

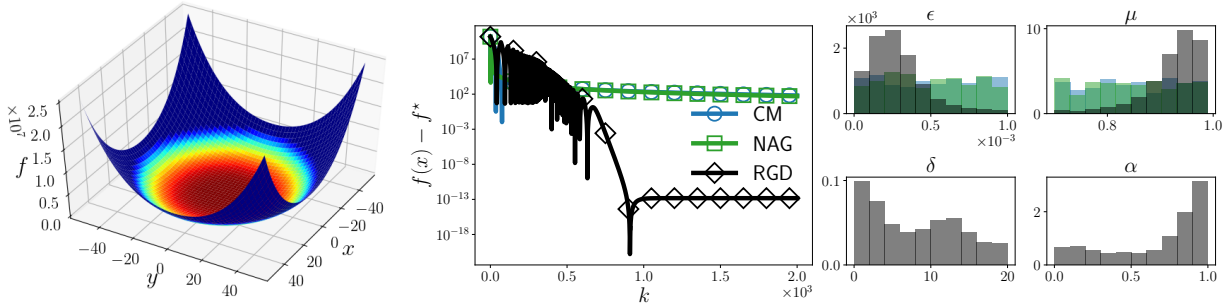


Figure 9: *Chung-Reynolds function*: $f(x) \equiv \left(\sum_{i=1}^{n} x_i^2\right)^2$. The global minimum is at $f(0) = 0$. This function is usually considered on the region $-100 \leq x_i \leq 100$. We consider $n = 50$ dimensions and initialize at $x_0 = (50, \ldots, 50)$. Note that RGD was able to improve convergence by controlling the kinetic energy with $\delta > 0$. We also see the benefits of being conformal symplectic, i.e. $\alpha \to 1$.

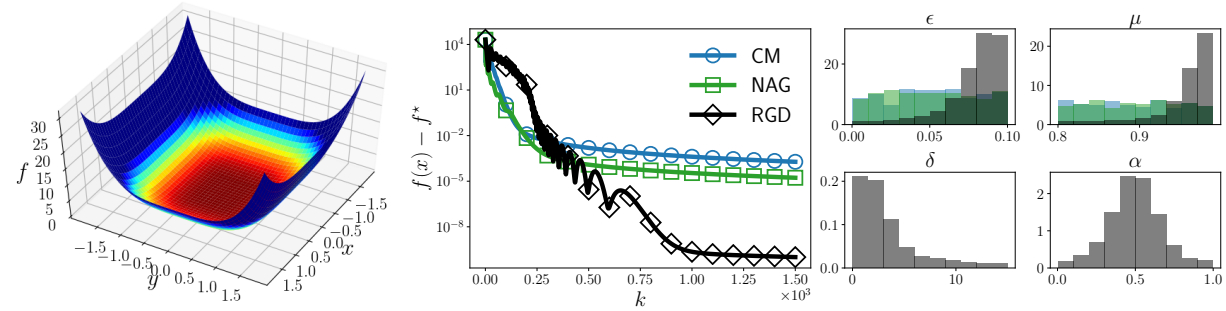


Figure 10: *Quartic function*: $f(x) \equiv \sum_{i=1}^{n} i x_i^4$. The global minimum is at $f(0) = 0$. This function is usually considered over $-1.28 \leq x_i \leq 1.28$. We choose $n = 50$ dimensions and initialize at $x_0 = (2, \ldots, 2)$.

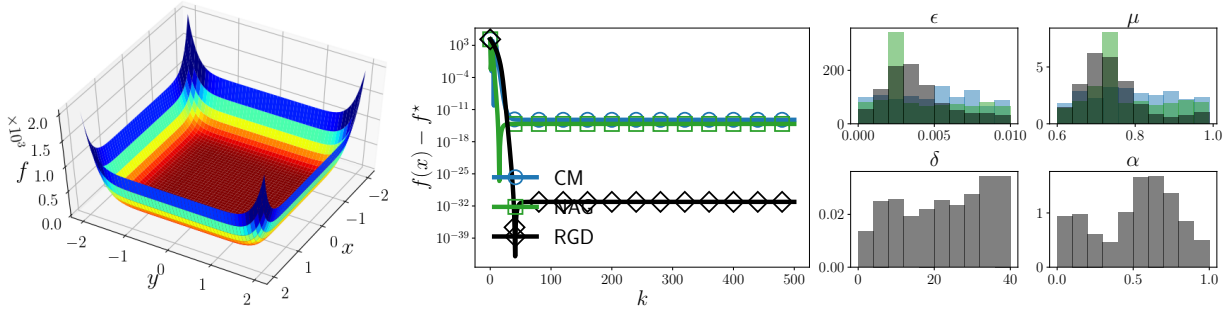


Figure 11: *Schwefel function*: $f(x) \equiv \sum_{i=1}^{n} x_i^{10}$. The minimum is at $f(0) = 0$. The function is usually considered over $-10 \leq x_i \leq 10$. This function grows even stronger than the previous two cases. We consider $n = 20$ dimensions and initialize at $x_0 = (2, \ldots, 2)$. Note that $\delta > 0$ is essential to control the kinetic energy and improve convergence.

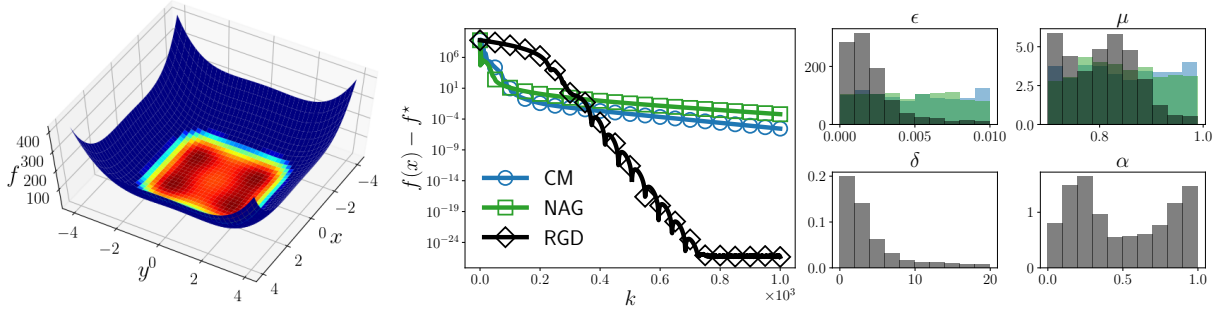


Figure 12: *Qing function*: $f(x) \equiv \sum_{i=1}^{n}(x_i^2 - i)^2$. This function is multimodal, with minimum at $x_i^\star = \pm\sqrt{i}$, $f(x^\star) = 0$. The function is usually studied in the region $-500 \leq x_i \leq 500$. We consider $n = 100$ dimensions with initialization at $x_0 = (50, \ldots, 50)$.

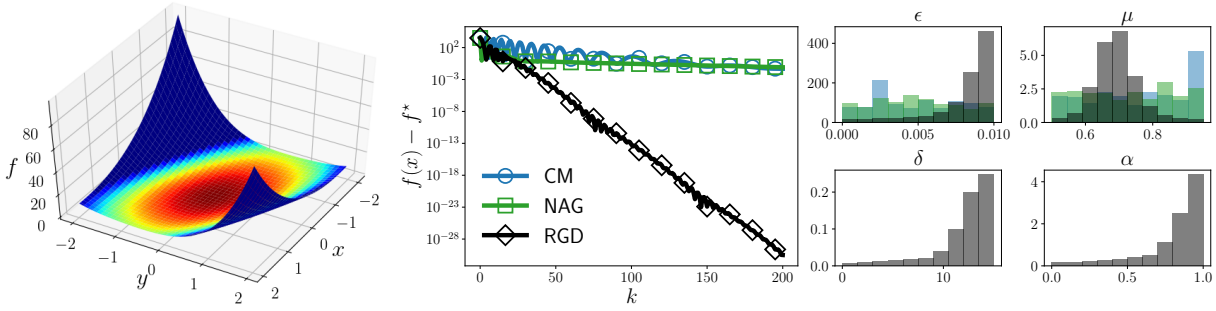


Figure 13: *Zakharov function*: $f(x) \equiv \sum_{i=1}^{n} x_i^2 + \left(\frac{1}{2}\sum_{i=1}^{n} i x_i\right)^2 + \left(\frac{1}{2}\sum_{i=1}^{n} i x_i\right)^4$. The minimum is at $f(0) = 0$. The region of interest is usually $-5 \leq x_i \leq 10$. We consider $n = 5$ and initialize at $x_0 = (1, \ldots, 1)$. Note that $\delta > 0$ played a dominant role here, and $\alpha \to 1$ as well. RGD successfully minimized this function to high accuracy, contrary to CM and NAG that were unable to get even close to the minimum.

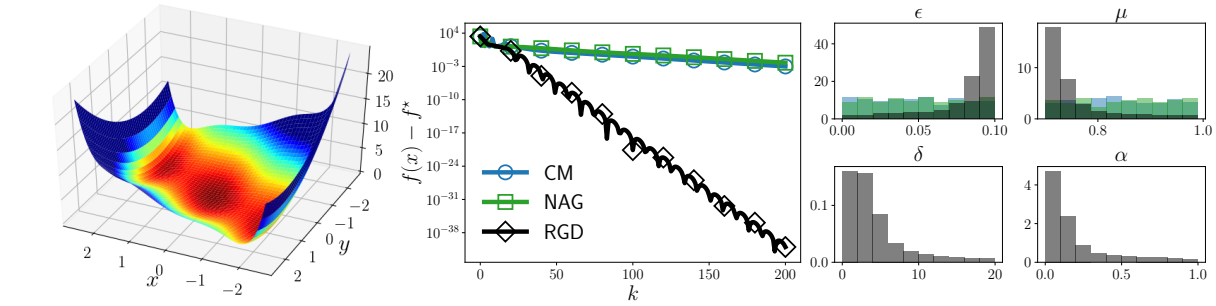


Figure 14: Three-hump camel back function: $f(x, y) \equiv 2x^2 - 1.05x^4 + x^6/6 + xy + y^2$. This is a multimodal function with global minimum is at $f(0, 0) = 0$. The region of interest is usually $-5 \leq x, y \leq 5$. We initialize at $x_0 = (5, 5)$. The two local minima are somewhat close to the global minimum which makes optimization challenging. Only RGD was able to minimize the function.

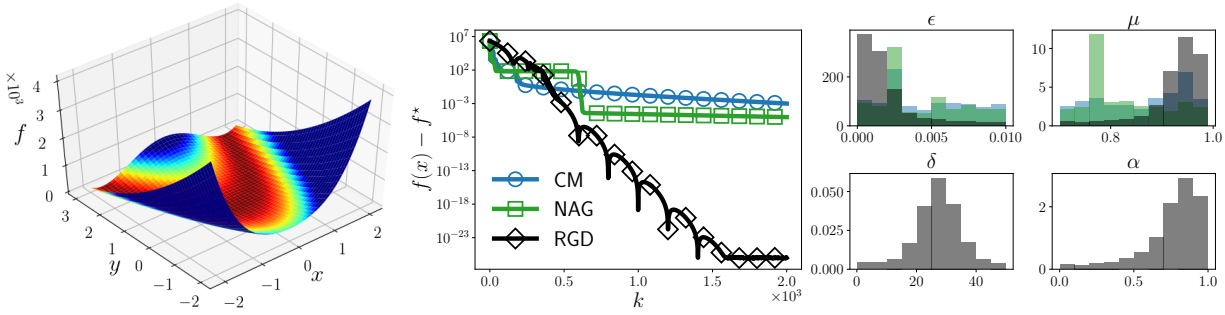


Figure 15: *Rosenbrock function*: $f(x) \equiv \sum_{i-1}^{n-1} \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right)$. The global minimum is at $f(1, \ldots, 1) = 0$. More details about this function was described in Section 7. Here we consider $n = 1000$ dimensions and initialize at $x_0 = (2.048, \ldots, 2.048)$. This function is usually studied in the region $-2.048 \leq x_i \leq 2.048$. Note that $\delta > 0$ was important for the improved convergence.

References

[1] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Comp. Math. and Math. Physics* **4** no. 5, (1964) 1–17.

[2] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$," *Dokl. Akad. Nauk SSSR* **269** (1983) 543–547.

[3] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Int. Conf. Machine Learning*. 2013.

[4] T. Tieleman and G. Hinton, "Lecture 6.5-RMSprop: Divide the gradient by a running average of its recent magnitude." Coursera: Neural Networks for Machine Learning, 2012.

[5] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Int. Conf. Learning Representations*. 2015.

[6] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods of online learning and stochastic optimization," *J. Machine Learning Research* **12** (2017) 2121–2159.

[7] T. Dozat, "Incorporating Nesterov momentum into Adam," in *Int. Conf. Learning Representations, Workshop*. 2016.

[8] W. Su, S. Boyd, and E. J. Candès, "A differential equation for modeling Nesterov's accelerated gradient method: theory and insights," *J. Machine Learning Research* **17** no. 153, (2016) 1–43.

[9] A. Wibisono, A. C. Wilson, and M. I. Jordan, "A variational perspective on accelerated methods in optimization," *Proc. Nat. Acad. Sci.* **113** no. 47, (2016) E7351–E7358.

[10] W. Krichene, A. Bayen, and P. L. Bartlett, "Accelerated mirror descent in continuous and discrete time," in *Advances in Neural Information Processing Systems*, vol. 28. 2015.

[11] J. Zhang, A. Mokhtari, S. Sra, and A. Jadbabaie, "Direct Runge-Kutta discretization achieves acceleration," in *Advances in Neural Information Processing Systems*, vol. 31. 2018.

[12] B. Shi, S. S. Du, M. I. Jordan, and W. J. Su, "Understanding the acceleration phenomenon via high-resolution differential equations." arXiv:1810.08907 [math.OC], 2018.

[13] L. F. Yang, R. Arora, V. Braverman, and T. Zhao, "The physical systems behind optimization algorithms," in *Advances on Neural Information Processing Systems*. 2018.

[14] M. Betancourt, M. I. Jordan, and A. C. Wilson, "On symplectic optimization," `arXiv:1802.03653 [stat.CO]`.

[15] G. França, D. P. Robinson, and R. Vidal, "ADMM and accelerated ADMM as continuous dynamical systems," *Int. Conf. Machine Learning* (2018) .

[16] G. França, D. P. Robinson, and R. Vidal, "A nonsmooth dynamical systems perspective on accelerated extensions of ADMM," `arXiv:1808.04048 [math.OC]`.

[17] G. França, D. P. Robinson, and R. Vidal, "Gradient flows and accelerated proximal splitting methods," `arXiv:1908.00865 [math.OC]`.

[18] G. França, M. I. Jordan, and R. Vidal, "On dissipative symplectic integration with applications to gradient-based optimization," `arXiv:2004.06840 [math.OC]`.

[19] R. I. McLachlan and G. R. W. Quispel, "Geometric integrators for ODEs," *J. Phys. A: Math. Gen.* **39** (2006) 5251–5285.

[20] E. Forest, "Geometric integration for particle accelerators," *J. Phys. A: Math. Gen.* **39** (2006) 5321—5377.

[21] P. J. Channell and C. Scovel, "Symplectic integration of Hamiltonian systems," *Nonlinearity* **3** (1990) 231–259.

[22] R. Quispel and R. McLachlan, "Geometric numerical integration of differential equations," *J. Phys. A: Math. Gen.* **39** (2006) .

[23] R. McLachlan and M. Perlmutter, "Conformal Hamiltonian systems," *Journal of Geometry and Physics* **39** (2001) 276–300.

[24] A. Bhatt, D. Floyd, and B. E. Moore, "Second order conformal symplectic schemes for damped Hamiltonian systems," *Journal of Scientific Computing* **66** (2016) 1234–1259.

[25] M. Muehlebach and M. I. Jordan, "Optimization with momentum: dynamical, control-theoretic, and symplectic perspectives," `arXiv:2002.12493 [math.OC]`.

[26] X. Lu, V. Perrone, L. Hasenclever, Y. W. Teh, and S. J. Vollmer, "Relativistic Monte Carlo," *20th Int. Conf. Articial Intelligence and Statistics* (2017) .

[27] S. Livingstone, M. F. Faulkner, and G. O. Roberts, "Kinetic energy choice in Hamiltonian/hybrid Monte Carlo," `arXiv:1706.02649 [stat.CO]`.

[28] C. J. Maddison, D. Paulin, Y. W. Teh, B. O'Donoghue, and A. Doucet, "Hamiltonian descent methods," `arXiv:1809.05042 [math.OC]`.

[29] H. Flanders, *Differential Forms with Applications to the Physical Sciences*. Dover, 1989.

[30] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration*. Springer, 2006.

[31] U. Dessler, "Symmetry property of the Lyapunov spectra of a class of dissipative dynamical systems with viscous damping," *Phys. Rev. A* **38** (1988) 2103.

[32] S. Marò and A. Sorrentino, "Aubry-Mather theory for conformally symplectic systems," *Commun. Math. Phys.* **354** (2017) 775–808.

[33] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," *Int. Conf. Machine Learning* (2013) .

[34] G. França, "Relativistic gradient descent (RGD)," 2020. `https://github.com/guisf/rgd.git`.

[35] H. H. Rosenbrock, "An automatic method for finding the greatest or least value of a function," *The Computer Journal* **3** no. 3, (1960) 175—-184.

[36] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[37] S. Kok and C. Sandrock, "Locating and characterizing the stationary points of the extended Rosenbrock function," *Evolutionary Computation* **17** no. 3, (2009) 437–453.

[38] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Trans. Information Theory* **56** (2009) .

[39] M. Jamil and X.-S. Yang, "A literature survey of benchmark functions for global optimization problems," *Int. J. Math. Modelling and Num. Optimisation* **4** no. 2, (2013) 150–194.