Entropy **2015**, *17*, 304-345; doi:10.3390/e17010304

OPEN ACCESS

entropy

ISSN 1099-4300

www.mdpi.com/journal/entropy

Article

Black-Box Optimization Using Geodesics in Statistical Manifolds[†]

Jérémy Bensadon

Laboratoire de Recherche en Informatique, Université Paris-Sud, 91400 Orsay, France;
E-Mail: jeremy.bensadon@u-psud.fr

[†] This paper is an extended version of our paper published in 34th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, 21–26 September 2014, Château Clos Lucé, Parc Leonardo Da Vinci, Amboise, France.

Received: 8 October 2014 / Accepted: 7 January 2015 / Published: 13 January 2015

Abstract: Information geometric optimization (IGO) is a general framework for stochastic optimization problems aiming at limiting the influence of arbitrary parametrization choices: the initial problem is transformed into the optimization of a smooth function on a Riemannian manifold, defining a parametrization-invariant first order differential equation and, thus, yielding an approximately parametrization-invariant algorithm (up to second order in the step size). We define the geodesic IGO update, a fully parametrization-invariant algorithm using the Riemannian structure, and we compute it for the manifold of Gaussians, thanks to Noether's theorem. However, in similar algorithms, such as CMA-ES (Covariance Matrix Adaptation - Evolution Strategy) and xNES (exponential Natural Evolution Strategy), the time steps for the mean and the covariance are decoupled. We suggest two ways of doing so: twisted geodesic IGO (GIGO) and blockwise GIGO. Finally, we show that while the xNES algorithm is not GIGO, it is an instance of blockwise GIGO applied to the mean and covariance matrix separately. Therefore, xNES has an almost parametrization-invariant description.

Keywords: black-box; optimization; geodesics; Gaussian; information geometry; natural gradient; Noether; learning rate; IGO; xNES

1. Introduction

Consider an objective function $f\colon X \to \mathbb{R}$ to be minimized. We suppose we have absolutely no knowledge about f: the only thing we can do is ask for its value at any point $x \in X$ (black-box optimization) and that the evaluation of f is a costly operation. We are going to study algorithms that can be described in the IGO framework (see [1]).

We consider the following optimization procedure:

We choose $(P_\theta)_{\theta\in\Theta}$ a family of probability distributions (which will be given a Riemannian manifold structure, following [2]) on X and an initial probability distribution P_{θ^0}. Now, we replace f by $F\colon \Theta \to \mathbb{R}$ (for example $F(\theta) = E_{x\sim P_\theta}[f(x)]$), and we optimize F by gradient descent, corresponding to the gradient flow:

$$\frac{\mathrm{d}\theta^t}{\mathrm{d}t} = -\nabla_\theta E_{x\sim P_\theta}[f(x)]. \tag{1}$$

However, because of the gradient, this equation depends entirely on the parametrization we chose for Θ, which is disturbing: we do not want to have two different updates, because we chose different parameters to represent the objects with which we are working. Moreover, in the case of a function with several local minima, changing the parametrization can change the attained optimum (see [3], for example). That is why invariance is a design principle behind IGO. More precisely, we want invariance with respect to monotone transformations of f and invariance under reparametrization of Θ.

The IGO framework uses the geometry of the family Θ, which is given by the Fisher metric to provide a differential equation on θ with the desired properties, but because of the discretization of time needed to obtain an explicit algorithm, we lose invariance under reparametrization of θ: two IGO algorithms applied to the same function to be optimized, but with different parametrizations, coincide only at first order in the step size. A possible solution to this problem is geodesic IGO (GIGO), introduced here (see also IGO-Maximum Likelihoodin [1], for example.): the initial direction of the update at each step of the algorithm remains the same as in IGO, but instead of moving straight for the chosen parametrization, we use the Riemannian manifold structure of our family of probability distributions (see [2]) by following its geodesics.

Finding the geodesics of a Riemannian manifold is not always easy, but Noether's theorem will allow us to obtain quantities that are preserved along the geodesics, thus allowing, in the case of Gaussian distributions, one to obtain a first order differential equation satisfied by the geodesics, which makes their computation easier.

Although the geodesic IGO algorithm is not, strictly speaking, parametrization-invariant when no closed form for the geodesics is known, it is possible to compute them at arbitrary precision without increasing the numbers of objective function calls.

The first two sections are preliminaries: in Section 2, we recall the IGO algorithm, introduced in [1], and in Section 3, after a reminder about Riemannian geometry, we state Noether's theorem, which will be our main tool to compute the GIGO update for Gaussian distributions.

In Section 4, we consider Gaussian distributions with a covariance matrix proportional to the identity matrix: this space is isometric to the hyperbolic space, and the geodesics of the latter are known.

In Section 5.1, we consider the general Gaussian case, and we use Noether's theorem to obtain two different sets of equations to compute the GIGO update. The equations are known (see [4–6]), but the connection with Noether's theorem has not been mentioned. We then give the explicit solution for these equations, from [5].

In Section 6, we recall quickly the xNES and CMA-ESupdates, and we introduce a slight modification of the IGO algorithm to incorporate the direction-dependent learning rates used in CMA-ESand xNES. We then compare these different algorithms and prove that xNES is not GIGO in general, and we finally introduce a new family of algorithms extending GIGO and recovering xNES from abstract principles.

Finally, Section 7 presents numerical experiments, which suggest that when using GIGO with Gaussian distributions, the step size must be chosen carefully.

2. Definitions: IGO, GIGO

In this section, we recall what the IGO framework is and we define the geodesic IGO update. Consider again Equation (1):

$$\frac{\mathrm{d}\theta^t}{\mathrm{d}t} = -\nabla_\theta E_{x\sim P_\theta}[f(x)].$$

As we saw in the Introduction:

- The gradient depends on the parametrization of our space of probability distributions (see Section 2.3 for an example).
- The equation is not invariant under monotone transformations of f. For example, the optimization for $10f$ moves ten times faster than the optimization for f.

In this section, we recall how IGO deals with this (see [1] for a better presentation).

2.1. Invariance under Reparametrization of θ: Fisher Metric

In order to achieve invariance under reparametrization of θ, it is possible to turn our family of probability distributions into a Riemannian manifold (this is the main topic of information geometry; see [2]), which allows us to use a canonical, parametrization-invariant gradient (called the natural gradient).

Definition 1. *Let P, Q be two probability distributions on X. The Kullback–Leibler divergence of Q from P is defined by:*

$$\mathrm{KL}(Q\|P) = \int_X \ln\left(\frac{Q(x)}{P(x)}\right)\mathrm{d}Q(x). \tag{2}$$

By definition, it does not depend on the parametrization. It is not symmetrical, but if for all x, the application $\theta \mapsto P_\theta(x)$ is C^2, then a second-order expansion yields:

$$\mathrm{KL}(P_{\theta+\mathrm{d}\theta}\|P_\theta) = \frac{1}{2}\sum_{i,j} I_{ij}(\theta)\mathrm{d}\theta_i\mathrm{d}\theta_j + o(\mathrm{d}\theta^2), \tag{3}$$

where:

$$I_{ij}(\theta) = \int_X \frac{\partial \ln P_\theta(x)}{\partial \theta_i} \frac{\partial \ln P_\theta(x)}{\partial \theta_j} \mathrm{d}P_\theta(x) = -\int_X \frac{\partial^2 \ln P_\theta(x)}{\partial \theta_i \partial \theta_j} \mathrm{d}P_\theta(x). \tag{4}$$

This is enough to endow the family $(P_\theta)_{\theta\in\Theta}$ with a Riemannian manifold structure: a Riemannian manifold M is a differentiable manifold, which can be seen as pieces of $\mathbb{R}^n$ glued together, with a metric. The metric at x is a symmetric positive-definite quadratic form on the tangent space of M at x: it indicates how expensive it is to move in a given direction on the manifold. We will think of the updates of the algorithms that we will be studying as paths on M.

The matrix $I(\theta)$ is called the "Fisher information matrix", and the metric it defines is called the "Fisher metric".

Given a metric, it is possible to define a gradient attached to this metric; the key property of the gradient is that for any smooth function f:

$$f(x+h) = f(x) + \sum_i h_i \frac{\partial f}{\partial x_i} + o(\|h\|) = f(x) + \langle h, \nabla f(x)\rangle + o(\|h\|), \tag{5}$$

where $\langle x, y\rangle = x^T I y$ is the dot product in metric I. Therefore, in order to keep the property of Equation (5), we must have $\nabla f = I^{-1}\frac{\partial f}{\partial x}$.

We have therefore the following gradient (called the "natural gradient"; see [2]):

$$\tilde{\nabla}_\theta = I^{-1}(\theta)\frac{\partial}{\partial \theta}, \tag{6}$$

and since the Kullback–Leibler divergence does not depend on the parametrization, neither does the natural gradient.

Later in this paper, we will study families of Gaussian distributions. The following proposition gives the Fisher metric for these families.

Proposition 1. *Let $(P_\theta)_{\theta\in\Theta}$ be a family of normal probability distributions: $P_\theta = \mathcal{N}(\mu(\theta), \Sigma(\theta))$. If μ and Σ are C^1, the Fisher metric is given by:*

$$I_{i,j}(\theta) = \frac{\partial \mu^T}{\partial \theta_i}\Sigma^{-1}\frac{\partial \mu}{\partial \theta_j} + \frac{1}{2}\operatorname{tr}\left(\Sigma^{-1}\frac{\partial \Sigma}{\partial \theta_i}\Sigma^{-1}\frac{\partial \Sigma}{\partial \theta_j}\right). \tag{7}$$

Proof. This is a non-trivial calculation. See [7] or [8] for more details. □

As we will often be working with Gaussian distributions, we introduce the following notation:

Notation 1. *$\mathbb{G}_d$ is the manifold of Gaussian distributions in dimension d, equipped with the Fisher metric. $\tilde{\mathbb{G}}_d$ is the manifold of Gaussian distributions in dimension d, with the covariance matrix proportional to identity in the canonical basis of $\mathbb{R}^d$, equipped with the Fisher metric.*

2.2. *IGO Flow, IGO Algorithm*

In IGO [1], invariance with respect to monotone transformations is achieved by replacing f by the following transform; we set:

$$q(x) = P_{x'\sim P_\theta}(f(x') \leqslant f(x)), \tag{8}$$

a non-increasing function $w\colon [0;1] \to \mathbb{R}$ is chosen (the selection scheme), and finally, $W_\theta^f(x) = w(q(x))$ (this definition has to be slightly changed if the probability of a tie is not zero, see [1] for more details). By performing a gradient descent on $E_{x\sim P_\theta}[W_{\theta^t}^f(x)]$, we obtain the "IGO flow":

$$\frac{\mathrm{d}\theta^t}{\mathrm{d}t} = \tilde{\nabla}_\theta \int_X W_{\theta^t}^f(x) P_\theta(\mathrm{d}x) = \int_X W_{\theta^t}^f(x) \tilde{\nabla}_\theta \ln P_\theta(x) P_{\theta^t}(\mathrm{d}x). \tag{9}$$

Notice that the function we are optimizing is $E_{x\sim P_\theta}[W_{\theta^t}^f(x)]$ and not $E_{x\sim P_\theta}[W_\theta^f(x)]$ (the second function is constant and always equal to $\int_0^1 w$). In particular, the function for which we are performing the gradient descent changes at each step, although their optimum (a Dirac at the minimum of f) does not: the IGO flow is not a gradient flow; it is simply a vector flow given by the gradient of interrelated functions.

For practical implementation, the integral in (9) has to be approximated. For the integral itself, the Monte-Carlo method is used; N values $(x_1, ..., x_N)$ are sampled from the distribution P_{θ^t}, and the integral becomes:

$$\frac{1}{N} \sum_{i=1}^N W_{\theta^t}^f(x_i) \tilde{\nabla}_\theta \ln P_\theta(x_i) \tag{10}$$

and we approximate $\frac{1}{N} W_\theta^f(x_i) = \frac{1}{N} w(q(x_i))$ by $\hat{w}_i = \frac{1}{N} w(\frac{\mathrm{rk}(x_i)+1/2}{N})$, where $\mathrm{rk}(x_i) = |\{j, f(x_j) < f(x_i)\}|$: it can be proven (see [1]) that $\lim_{N\to\infty} N\hat{w}_i = W_f^{\theta^t}(x_i)$ (here again, we are assuming that there are no ties).

We now have an algorithm that can be used in practice if the Fisher information matrix is known.

Definition 1. *The IGO update associated with parametrization θ, sample size N, step size δt and selection scheme w is given by the following update rule:*

$$\theta^{t+\delta t} = \theta^t + \delta t I^{-1}(\theta^t) \sum_{i=1}^N \hat{w}_i \frac{\partial \ln P_\theta(x_i)}{\partial \theta}. \tag{11}$$

We call IGO speed the vector $I^{-1}(\theta^t) \sum_{i=1}^N \hat{w}_i \frac{\partial \ln P_\theta(x_i)}{\partial \theta}$.

Notice that one could start directly with the $\hat{w}_i$ rather than w, as we will do later.

Replacing f by its expected value under a probability distribution P_θ and optimizing over θ using the natural gradient have already been discussed. For example, in the case of a function f defined on $\{0,1\}^n$, IGO with the Bernoulli distributions yields the algorithm, PBIL[9]. Another similar approach (stochastic relaxation) is given in [10]. For a continuous function, as we will see later, the IGO framework recovers several known ranked-based natural gradient algorithms, such as pure rank-μ CMA-ES [11], xNES or SNES (Separable Natural Evolution Strategies) [12]. See [13] or [14] for other, not necessarily gradient-based, optimization algorithms on manifolds.

2.3. Geodesic IGO

Although the IGO flow associated with a family of probability distributions is intrinsic (it only depends on the family itself, not the parametrization we choose for it), the IGO update is not. However, the difference between two steps of IGO that differ only by the parametrization is only $O(\delta t^2)$, whereas the different between two vanilla gradient descents with different parametrizations is $O(\delta t)$.

Intuitively, the reason for this difference is that two IGO algorithms start at the same point and follow "straight lines" with the same initial speed, but the definition of "straight lines" changes with the parametrization.

For instance, in the case of Gaussian distributions, let us consider two different IGO updates with Gaussian distributions in dimension one, the first one with parametrization (μ, σ) and the second one with parametrization $(\mu, c := \sigma^2)$. We suppose that the IGO speed for the first algorithm is $(\dot{\mu}, \dot{\sigma})$. The corresponding IGO speed in the second parametrization is given by the identity $\dot{c} = 2\sigma\dot{\sigma}$. Therefore, the first algorithm gives the standard deviation $\sigma_{\text{new},1} = \sigma_{\text{old}} + \delta t\dot{\sigma}$ and the variance $c_{\text{new},1} = (\sigma_{\text{new},1})^2 = c_{\text{old}} + 2\delta t\sigma_{\text{old}}\dot{\sigma} + \delta t^2\dot{\sigma}^2 = c_{\text{new},2} + \delta t^2\dot{\sigma}^2$.

The geodesics of a Riemannian manifold are the generalization of the notion of a straight line: they are curves that locally minimize length. In particular, given two points a and b on the Riemannian manifold M, the shortest path from a to b is always a geodesic (the converse is not true, though). The notion will be explained precisely in Section 3, but let us define the geodesic IGO algorithm, which follows the geodesics of the manifold instead of following the straight lines for an arbitrary parametrization.

Definition 2 (GIGO)**.** *The geodesic IGO update (GIGO) associated with sample size N, step size δt and selection scheme w is given by the following update rule:*

$$\theta^{t+\delta t} = \exp_{\theta^t}(Y\delta t) \tag{12}$$

where:

$$Y = I^{-1}(\theta^t)\sum_{i=1}^{N}\hat{w}_i\frac{\partial \ln P_\theta(x_i)}{\partial\theta}, \tag{13}$$

is the IGO speed and $\exp_{\theta^t}$ is the exponential of the Riemannian manifold Θ. Namely, $\exp_{\theta^t}(Y\delta t)$ is the endpoint of the geodesic of Θ starting at θ^t, with initial speed Y, after a time δt. By definition, this update does not depend on the parametrization θ.

Notice that while the GIGO update is compatible with the IGO flow (in the sense that when $\delta t \to 0$ and $N \to \infty$, a parameter θ^t updated according to the GIGO algorithm is a solution of Equation (9), the equation defining the IGO flow), it not necessarily an IGO update. More precisely, the GIGO update is an IGO update if and only if the geodesics of Θ are straight lines for some parametrization (by Beltrami's theorem, this is equivalent to Θ having constant curvature).

This is a particular case of a retraction [14]: a map from the tangent bundle of a manifold to the manifold itself satisfying a rigidity condition. Arguably, the Riemannian exponential is the most natural retraction, since it depends only on the Riemannian manifold itself and not on any decomposition. However, in general, the geodesics are difficult to compute.

In the next section, we will state Noether's theorem, which will be our main tool to compute the GIGO update for Gaussian distributions.

3. Riemannian Geometry, Noether's Theorem

3.1. Riemannian Geometry

The goal of this section is to state Noether's theorem. See [15] for the proofs and [16] or [17] for a more detailed presentation. Noether's theorem states that if a system has symmetries, then there are invariants attached to these symmetries. Firstly, we need some definitions.

Definition 3 (Motion in a Lagrangian system)**.** *Let M be a differentiable manifold, TM the set of tangent vectors on M (a tangent vector is identified by the point at which it is tangent and a vector in the tangent space) and* $\begin{array}{ccccc} \mathcal{L} & : & TM & \rightarrow & \mathbb{R} \\ & & (q,v) & \mapsto & \mathcal{L}(q,v) \end{array}$ *a differentiable function called the Lagrangian function (in general, it could depend on t). A "motion in the Lagrangian system $(M, \mathcal{L})$ from x to y" is map $\gamma\colon [t_0, t_1] \to M$, such that:*

- $\gamma(t_0) = x$
- $\gamma(t_1) = y$
- *γ is a local extremum of the functional:*

$$\Phi(\gamma) = \int_{t_0}^{t_1} \mathcal{L}(\gamma(t), \dot{\gamma}(t))\mathrm{d}t, \tag{14}$$

among all curves $c\colon [t_0, t_1] \to M$, such that $c(t_0) = x$, and $c(t_1) = y$.

For example, when (M, g) is a Riemannian manifold, the length of a curve γ between $\gamma(t_0)$ and $\gamma(t_1)$ is:

$$\int_{t_0}^{t_1} \sqrt{g(\dot{\gamma}(t), \dot{\gamma}(t))}\mathrm{d}t. \tag{15}$$

The curves that follow the shortest path between two points $x, y \in M$ are therefore the minima γ of the functional (15), such that $\gamma(t_0) = x$ and $\gamma(t_1) = y$, and the corresponding Lagrangian function is $(q, v) \mapsto \sqrt{g(v, v)}$. However, any curve following the shortest trajectory will have minimum length. For example, if $\gamma_1 : [a, b] \to M$ is a curve of the shortest path, so is $\gamma_2 : t \mapsto \gamma_1(t^2)$: these two curves define the same trajectory in M, but they do not travel along this trajectory at the same speed. This leads us to the following definition:

Definition 4 (Geodesics)**.** *Let I be an interval of $\mathbb{R}$ and (M, g) be a Riemannian manifold. A curve $\gamma : I \to M$ is called a geodesic if for all $t_0, t_1 \in I$, $\gamma|_{[t_0,t_1]}$ is a motion in the Lagrangian system $(M, \mathcal{L})$ from $\gamma(t_0)$ to $\gamma(t_1)$, where:*

$$\mathcal{L}(\gamma) = \int_{t_0}^{t_1} g(\dot{\gamma}(t), \dot{\gamma}(t))\mathrm{d}t. \tag{16}$$

It can be shown (see [16]) that geodesics are curves that locally minimize length, with constant velocity, in the sense that $\frac{dg(\dot{\gamma}(t),(\dot{\gamma}(t))}{dt} = 0$. In particular, given a starting point and a starting speed, the geodesic is unique. This motivates the definition of the exponential of a Riemannian manifold.

Definition 5. *Let (M, g) be a Riemannian manifold. We call the exponential of M the application:*

$$\begin{array}{rcccc} \exp & : & TM & \to & M \\ & & (x, v) & \mapsto & \exp_x(v), \end{array}$$

such that for any $x \in M$, if γ is the geodesic of M satisfying $\gamma(0) = x$ and $\gamma'(0) = v$, then $\exp_x(v) = \gamma(1)$.

In order to find an extremal of a functional, the most commonly-used result is called the "Euler–Lagrange equations" (see [15], for example); a motion γ in the Lagrangian system $(M, \mathcal{L})$ must satisfy:

$$\frac{\partial \mathcal{L}}{\partial x}(\gamma(t)) - \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}(\dot{\gamma}(t))\right) = 0. \tag{17}$$

By applying this equation with the Lagrangian given by (16), it is possible to show that the geodesics of a Riemannian manifold follow the "geodesic equations":

$$\ddot{x}^k + \Gamma^k_{ij}\dot{x}^i\dot{x}^j = 0, \tag{18}$$

where the

$$\Gamma^k_{ij} = \frac{1}{2}g^{lk}\left(\frac{\partial g_{jl}}{\partial q_i} + \frac{\partial g_{li}}{\partial q_j} - \frac{\partial g_{ij}}{\partial q_l}\right) \tag{19}$$

are called "Christoffel symbols" of the metric g. However, these coefficients are tedious (and sometimes difficult) to compute, and (18) is a second order differential equation. Noether's theorem will give us a first order equation to compute the geodesics.

3.2. Noether's Theorem

Definition 6. *Let $h\colon M \to M$, a diffeomorphism. We say that the Lagrangian system $(M, \mathcal{L})$ admits the symmetry h if for any $(q, v) \in TM$,*

$$\mathcal{L}\left(h(q), \mathrm{d}h(v)\right) = \mathcal{L}(q, v), \tag{20}$$

where $\mathrm{d}h$ is the differential of h.

If M is clear in the context, we will sometimes say that $\mathcal{L}$ is invariant under h.

An example will be given in the proof of Theorem 3.

We can now state Noether's theorem (see, for example, [15]).

Theorem 1 (Noether's Theorem)**.** *If the Lagrangian system $(M, \mathcal{L})$ admits the one-parameter group of symmetries $h^s\colon M \to M$, $s \in \mathbb{R}$, then the following quantity remains constant during motions in the system $(M, \mathcal{L})$. Namely,*

$$I(\gamma(t), \dot{\gamma}(t)) = \frac{\partial \mathcal{L}}{\partial v}\left(\frac{\mathrm{d}h^s(\gamma(t))}{\mathrm{d}s}|_{s=0}\right) \tag{21}$$

does not depend on t if γ is a motion in $(M, \mathcal{L})$.

Now, we are going to apply this theorem to our problem: computing the geodesics of Riemannian manifolds of Gaussian distributions.

4. GIGO in $\tilde{\mathbb{G}}_d$

If we force the covariance matrix to be either diagonal or proportional to the identity matrix, the geodesics have a simple expression that we give below. In the former case, the manifold we are considering is $(\mathbb{G}_1)^d$, and in the latter case, it is $\tilde{\mathbb{G}}_d$.

The geodesics of $(\mathbb{G}_1)^d$ are given by:

Proposition 2. *Let M be a Riemannian manifold; let $d \in \mathbb{N}$; let Φ be the Riemannian exponential of M^d; and let ϕ be the Riemannian exponential of M. We have:*

$$\Phi_{(x_1,...,x_n)}((v_1,...,v_n)) = (\phi_{x_1}(v_1),...,\phi_{x_n}(v_n)) \tag{22}$$

In particular, knowing the geodesics of $\mathbb{G}_1$ is enough to compute the geodesics of $(\mathbb{G}_1)^d$.

This is true, because a block of the product metric does not depend on variables of the other blocks.

Consequently, a GIGO update with a diagonal covariance matrix with the sample (x_i) is equivalent to d separate one-dimensional GIGO updates using the same samples. Moreover, $\mathbb{G}_1 \cong \tilde{\mathbb{G}}_1$, the geodesics of which are given below.

We will show that $\tilde{\mathbb{G}}_d$ and the "hyperbolic space", of which the geodesics are known, are isometric.

4.1. Preliminaries: Poincaré Half-Plane, Hyperbolic Space

In dimension two, the hyperbolic space is called the "hyperbolic plane" or the Poincaré half-plane. We recall its definition:

Definition 7 (Poincaré half-plane)**.** *We call the "Poincaré half-plane" the Riemannian manifold:*

$$\mathcal{H} = \{(x,y) \in \mathbb{R}^2,\, y > 0\},$$

with the metric $ds^2 = \frac{dx^2+dy^2}{y^2}$.

We also recall the expression of its geodesics (see, for example, [18]):

Proposition 3 (Geodesics of the Poincaré half-plane)**.** *The geodesics of the Poincaré half-plane are exactly the:*

$$t \mapsto \left(\mathrm{Re}(z(t)), \mathrm{Im}(z(t))\right),$$

where:

$$z(t) = \frac{aie^{vt} + b}{cie^{vt} + d}, \tag{23}$$

with $ad - bc = 1$ and $v > 0$.

The geodesics are half-circles perpendicular to the line $y = 0$ and vertical lines, as shown in Figure 1 below.

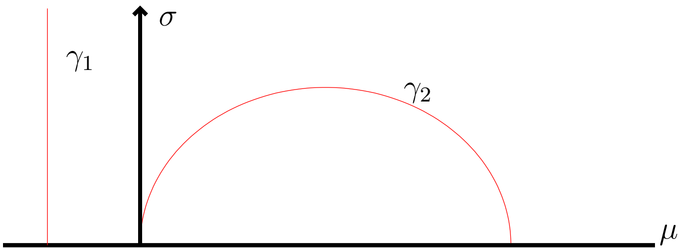

Figure 1. Geodesics of the Poincaré half-plane.

The generalization to the higher dimension is the following:

Definition 8 (Hyperbolic space)**.** *We call the "hyperbolic space of dimension n" the Riemannian manifold:*

$$\mathcal{H}_n = \{(x_1, ..., x_{n-1}, y) \in \mathbb{R}^n, y > 0\},$$

with the metric $ds^2 = \frac{dx_1^2 + ... + dx_{n-1}^2 + dy^2}{y^2}$ (or equivalently, the metric given by the matrix $\mathrm{Diag}(\frac{1}{y^2})$).

The Lagrangian for the geodesics is invariant under all translations along the x_i, so by Noether's theorem, its geodesics stay in a plane containing the direction y and the initial speed . The induced metric on this plane is the metric of the Poincaré half-plane. The geodesics are therefore given by the following proposition:

Proposition 4 (Geodesics of the hyperbolic space)**.** *If $\gamma\colon t \mapsto (x_1(t), ..., x_{n-1}(t), y(t)) = (\boldsymbol{x}(t), y(t))$ is a geodesic of $\mathcal{H}_n$, then there exists $a, b, c, d \in \mathbb{R}$, such that $ad - bc = 1$, and $v > 0$, such that $\boldsymbol{x}(t) = \boldsymbol{x}(0) + \frac{\dot{\boldsymbol{x}}_0}{\|\dot{\boldsymbol{x}}_0\|}\tilde{x}(t)$, $y(t) = \mathrm{Im}(\gamma_{\mathbb{C}}(t))$, with $\tilde{x}(t) = \mathrm{Re}(\gamma_{\mathbb{C}}(t))$ and:*

$$\gamma_{\mathbb{C}}(t) := \frac{aie^{vt} + b}{cie^{vt} + d}. \tag{24}$$

4.2. Computing the GIGO Update in $\tilde{\mathbb{G}}_d$

If we want to implement the GIGO algorithm in $\tilde{\mathbb{G}}_d$, we need to compute the natural gradient in $\tilde{\mathbb{G}}_d$ and to be able to compute the Riemannian exponential of $\tilde{\mathbb{G}}_d$.

Using Proposition 1, we can compute the metric of $\tilde{\mathbb{G}}_d$ in the parametrization $(\mu, \sigma) \mapsto \mathcal{N}(\mu, \sigma^2 I)$. We find:

$$\begin{pmatrix} \frac{1}{\sigma^2} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \frac{1}{\sigma^2} & 0 \\ 0 & \dots & 0 & \frac{2d}{\sigma^2} \end{pmatrix}. \tag{25}$$

Since this matrix is diagonal, it is easy to invert, and we immediately have the natural gradient and, consequently, the IGO speed.

Proposition 5. *In $\tilde{\mathbb{G}}_d$, the IGO speed Y is given by:*

$$Y_\mu = \sum_i \hat{w}_i (x_i - \mu), \tag{26}$$

$$Y_\sigma = \sum_i \hat{w}_i \left(\frac{(x_i - \mu)^T (x_i - \mu)}{2d\sigma} - \frac{\sigma}{2} \right). \tag{27}$$

Proof. We recall the IGO speed is defined by $Y = I^{-1}(\theta^t) \sum_{i=1}^N \hat{w}_i \frac{\partial \ln P_\theta(x_i)}{\partial \theta}$. Since $P_{\mu,\sigma}(x) = (2\pi\sigma^2)^{-d/2} \exp(-\frac{(x-\mu)^T(x-\mu)}{2\sigma^2})$, we have:

$$\frac{\partial \ln P_{\mu,\sigma}(x)}{\partial \mu} = x - \mu,$$

$$\frac{\partial \ln P_{\mu,\sigma}(x)}{\partial \sigma} = -\frac{d}{\sigma} + \frac{(x - \mu)^T (x - \mu)}{\sigma^3}.$$

The result follows. □

The metric defined by Equation (25) is not exactly the metric of the hyperbolic space, but with the substitution $\mu \leftarrow \frac{\mu}{\sqrt{2d}}$, the metric becomes $\frac{2d}{\sigma^2} I$, which is proportional to the metric of the hyperbolic space and, therefore, defines the same geodesics.

Theorem 2 (Geodesics of $\tilde{\mathbb{G}}_d$)**.** *If $\gamma \colon t \mapsto \mathcal{N}(\mu(t), \sigma(t)^2 I)$ is a geodesic of $\tilde{\mathbb{G}}_d$, then there exists $a, b, c, d \in \mathbb{R}$, such that $ad - bc = 1$, and $v > 0$, such that:*
$\mu(t) = \mu(0) + \sqrt{2d} \frac{\dot{\mu}_0}{\|\dot{\mu}_0\|} \tilde{r}(t)$, $\sigma(t) = \operatorname{Im}(\gamma_{\mathbb{C}}(t))$, with $\tilde{r}(t) = \operatorname{Re}(\gamma_{\mathbb{C}}(t))$ and

$$\gamma_{\mathbb{C}}(t) := \frac{aie^{vt} + b}{cie^{vt} + d}. \tag{28}$$

Now, in order to implement the corresponding GIGO algorithm, we only need to be able to find the coefficients a, b, c, d, v corresponding to an initial position (μ_0, σ_0) and an initial speed $(\dot{\mu}_0, \dot{\sigma}_0)$. This is a tedious but easy computation, the result of which is given in Proposition 17.

The pseudocode of GIGO in $\tilde{\mathbb{G}}_d$ is also given in the Appendix: it is obtained by concatenating Algorithms 1 and 7 (Proposition 17 and the pseudocode in the Appendix allow the metric to be slightly modified; see Section 6.2).

5. GIGO in $\mathbb{G}_d$

5.1. Obtaining a First Order Differential Equation for the Geodesics of $\mathbb{G}_d$

In the case where both the covariance matrix and the mean can vary freely, the equations of the geodesics have been computed in [4] and [5]. However, these articles start with the equations of the geodesics obtained with the Christoffel symbols, then partially integrate them . These equations are in fact a consequence of Noether's theorem and can be found directly.

Theorem 3. *Let $\gamma : t \mapsto \mathcal{N}(\mu_t, \Sigma_t)$ be a geodesic of $\mathbb{G}_d$. Then, the following quantities do not depend on t:*

$$J_\mu = \Sigma_t^{-1} \dot{\mu}_t, \tag{29}$$

$$J_\Sigma = \Sigma_t^{-1}(\dot{\mu}_t \mu_t^T + \dot{\Sigma}_t). \tag{30}$$

Proof. This is a direct application of Noether's theorem, with suitable groups of diffeomorphisms. By Proposition 1, the Lagrangian associated with the geodesics of $\mathbb{G}_d$ is:

$$\mathcal{L}(\mu, \Sigma, \dot{\mu}, \dot{\Sigma}) = \dot{\mu}^T \Sigma^{-1} \dot{\mu} + \frac{1}{2} \operatorname{tr}(\dot{\Sigma}\Sigma^{-1}\dot{\Sigma}\Sigma^{-1}). \tag{31}$$

Its derivative is:

$$\frac{\partial \mathcal{L}}{\partial \dot{\theta}} = \left[(h, H) \mapsto 2\dot{\mu}^T \Sigma^{-1} h + \operatorname{tr}(H\Sigma^{-1}\dot{\Sigma}\Sigma^{-1})\right]. \tag{32}$$

Let us show that this Lagrangian is invariant under affine changes of basis (thus illustrating Definition 6).

The general form of an affine change of basis is $\phi_{\mu_0,A} : (\mu, \Sigma) \mapsto (A\mu + \mu_0, A\Sigma A^T)$, with $\mu_0 \in \mathbb{R}^d$ and $A \in \mathrm{GL}_d(\mathbb{R})$.

We have:

$$\mathcal{L}(\phi_{\mu_0,A}(\mu,\Sigma), d\phi_{\mu_0,A}(\dot{\mu},\dot{\Sigma})) = \dot{\overline{A\mu}}^T (A\Sigma A^T)^{-1} \dot{\overline{A\mu}} + \frac{1}{2} \operatorname{tr}\left(\dot{\overline{A\Sigma A^T}}(A\Sigma A^T)^{-1}\dot{\overline{A\Sigma A^T}}(A\Sigma A^T)^{-1}\right), \tag{33}$$

and since $\dot{\overline{A\mu}} = A\dot{\mu}$ and $\dot{\overline{A\Sigma A^T}} = A\dot{\Sigma}A^T$, we find easily that:

$$\mathcal{L}(\phi_{\mu_0,A}(\mu,\Sigma), d\phi_{\mu_0,A}(\dot{\mu},\dot{\Sigma})) = \mathcal{L}(\mu, \Sigma, \dot{\mu}, \dot{\Sigma}), \tag{34}$$

or in other words: $\mathcal{L}$ is invariant under $\phi_{\mu_0,A}$ for any $\mu_0 \in \mathbb{R}^d$, $A \in GL_d(\mathbb{R})$.

In order to use Noether's theorem, we also need one-parameter groups of transformations. We choose the following:

(1) Translations of the mean vector. For any $i \in [1, d]$, let $h_i^s \colon (\mu, \Sigma) \mapsto (\mu + se_i, \Sigma)$, where e_i is the i-th basis vector. We have $\frac{dh_i^s}{ds}|_{s=0} = (e_i, 0)$, so by Noether's theorem,

$$\frac{\partial \mathcal{L}}{\partial \dot{\theta}}(e_i, 0) = 2\dot{\mu}^T \Sigma^{-1} e_i = 2e_i^T \Sigma^{-1} \dot{\mu}$$

remains constant for all i. The fact that J_μ is an invariant immediately follows.

(2) Linear base changes. For any $i, j \in [1, d]$, let $h_{i,j}^s : (\mu, \Sigma) \mapsto (\exp(sE_{ij})\mu, \exp(sE_{ij})\Sigma\exp(sE_{ji}))$, where E_{ij} is the matrix with a one at position (i, j) and zeros elsewhere. We have:

$$\frac{\mathrm{d}h_{E_{ij}}^s}{\mathrm{d}s}|_{s=0} = (E_{ij}\mu, E_{ij}\Sigma + \Sigma E_{ji}).$$

Therefore, by Noether's theorem, we then obtain the following invariants:

$$J_{ij} := \frac{\partial \mathcal{L}}{\partial \dot{\theta}}(E_{ij}\mu, E_{ij}\Sigma + \Sigma E_{ji}) \tag{35}$$

$$= 2\,\dot{\mu}^T \Sigma^{-1} E_{ij}\mu + \operatorname{tr}((E_{ij}\Sigma + \Sigma E_{ji})\Sigma^{-1}\dot{\Sigma}\Sigma^{-1}) \tag{36}$$

$$= 2\,(\Sigma^{-1}\dot{\mu})^T E_{ij}\mu + \operatorname{tr}(E_{ij}\dot{\Sigma}\Sigma^{-1}) + \operatorname{tr}(E_{ji}\Sigma^{-1}\dot{\Sigma}) \tag{37}$$

$$= 2(J_\mu \mu^T)_{ij} + 2(\Sigma^{-1}\dot{\Sigma})_{ij}, \tag{38}$$

and the coefficients of J_Σ in (30) are the $(J_{ij}/2)$.

□

This leads us to first order equations satisfied by the geodesics mentioned in [4–6].

Theorem 4 (GIGO-Σ)**.** *$t \mapsto \mathcal{N}(\mu_t, \Sigma_t)$ is a geodesic of $\mathbb{G}_d$ if and only if $\mu : t \mapsto \mu_t$ and $\Sigma : t \mapsto \Sigma_t$ satisfy the equations:*

$$\dot{\mu}_t = \Sigma_t J_\mu \tag{39}$$

$$\dot{\Sigma}_t = \Sigma_t (J_\Sigma - J_\mu \mu_t^T) = \Sigma_t J_\Sigma - \dot{\mu}_t \mu_t^T, \tag{40}$$

where:

$$J_\mu = \Sigma_0^{-1} \dot{\mu}_0,$$

and:

$$J_\Sigma = \Sigma_0^{-1} \left(\dot{\mu}_0 \mu_0^T + \dot{\Sigma}_0 \right).$$

Proof. This is an immediate consequence of Proposition 3. □

These equations can be solved analytically (see [5]); however, usually, that is not the case, and they have to be solved numerically, for example with the Euler method (the corresponding algorithm, which we call GIGO-Σ, is described in the Appendix). The goal of the remainder of the subsection is to show that having to use the Euler method is fine.

To avoid confusion, we will call the step size of the GIGO algorithm (δt in Proposition 2) "GIGO step size" and the step size of the Euler method (inside a step of the GIGO algorithm) "Euler step size".

Having to solve our equations numerically brings two problems:

The first one is a theoretical problem: the main reason to study GIGO is its invariance under reparametrization of θ, and we lose this invariance property when we use the Euler method. However, GIGO can get arbitrarily close to invariance by decreasing the Euler step size. In other words, the difference between two different IGO algorithms is $O(\delta t^2)$, and the difference between two different implementations of the GIGO algorithm is $O(h^2)$, where h is the Euler step size; it is easier to reduce the latter. Still, without a closed form for the geodesics of $\mathbb{G}_d$, the GIGO update is rather expensive to compute, but it can be argued that most of the computation time will still be the computation of the objective function f.

The second problem is purely numerical: we cannot guarantee that the covariance matrix remains positive-definite along the Euler method. Here, apart from finding a closed form for the geodesics, we have two solutions.

We can enforce this *a posteriori*: if the covariance matrix we find is not positive-definite after a GIGO step, we repeat the failed GIGO step with a reduced Euler step size (in our implementation, we divided it by four; see Algorithm 2 in the Appendix.).

The other solution is to obtain differential equations on a square root of the covariance matrix (*any* matrix A, such that $\Sigma = AA^T$).

Theorem 5 (GIGO-A)**.** *If $\mu : t \mapsto \mu_t$ and $A : t \mapsto A_t$ satisfy the equations:*

$$\dot{\mu}_t = A_t A_t^T J_\mu, \tag{41}$$

$$\dot{A}_t = \frac{1}{2} (J_\Sigma - J_\mu \mu_t^T)^T A_t, \tag{42}$$

where:

$$J_\mu = (A_0^{-1})^T A_0^{-1} \mu_0$$

and:

$$J_\Sigma = (A_0^{-1})^T A_0^{-1} (\dot{\mu}_0 \mu_0^T + \dot{A}_0 A_0^T + A_0 \dot{A}_0^T),$$

then $t \mapsto \mathcal{N}(\mu_t, A_t A_t^T)$ *is a geodesic of* $\mathbb{G}_d$*.*

Proof. This is a simple rewriting of Theorem 4: if we write $\Sigma := AA^T$, we find that J_μ and J_Σ are the same as in Theorem 4, and we have:

$$\dot{\mu} = \Sigma J_\mu,$$

and:

$$\dot{\Sigma} = (\dot{A}A^T + A\dot{A}^T) = \frac{1}{2}(J_\Sigma - J_\mu \mu^T)^T AA^T + \frac{1}{2} AA^T (J_\Sigma - J_\mu \mu^T)$$
$$= \frac{1}{2}(J_\Sigma - J_\mu \mu^T)^T \Sigma + \frac{1}{2}\Sigma(J_\Sigma - J_\mu \mu^T) = \frac{1}{2}\Sigma(J_\Sigma - J_\mu \mu^T) + \frac{1}{2}[\Sigma(J_\Sigma - J_\mu \mu^T)]^T.$$

By Theorem 4, $\Sigma(J_\Sigma - J_\mu \mu^T)$ is symmetric (since $\dot{\Sigma}$ has to be symmetric). Therefore, we have $\dot{\Sigma} = \Sigma(J_\Sigma - J_\mu \mu^T)$, and the result follows. □

Notice that Theorem 5 gives an equivalence, whereas Theorem 4 does not. The reason is that the square root of a symmetric positive-definite matrix is not unique. Still, it is canonical; see the discussion in Section 6.1.2.

As for Theorem 4, we can solve Equations (41) and (42) numerically, and we obtain another algorithm (Algorithm 3 in the Appendix; we will call it GIGO-A), with a behavior similar to the previous one (with Equations (39) and (40)). For both of them, numerical problems can arise when the covariance matrix is almost singular.

We have not managed to find any example where one of these two algorithms converged to the minimum of the objective function, whereas the other did not, and their behavior is almost the same.

More interestingly, the performances of these two algorithms are also the same as the performances of the exact GIGO algorithm, using the equations of Section 5.2.

Notice that even though GIGO-A directly maintains a square root of the covariance matrix, which makes sampling new points easier (to sample a point from $\mathcal{N}(\mu, \Sigma)$, a square root of Σ is needed), both GIGO-Σ and GIGO-A still have to invert the covariance matrix (or its square root) at each step, which is as costly as the decomposition, so one of these algorithms is roughly as expensive to compute as the other.

5.2. Explicit Form of the Geodesics of $\mathbb{G}_d$ *(from [5])*

We now give the exact geodesics of $\mathbb{G}_d$: the following results are a rewriting of Theorem 3.1 and its first corollary in [5].

Theorem 6. *Let* $(\dot{\mu}_0, \dot{\Sigma}_0) \in T_{\mathcal{N}(0,I)}\mathbb{G}_d$*. The geodesic of* $\mathbb{G}_d$ *starting from* $\mathcal{N}(0,1)$ *with initial speed* $(\dot{\mu}_0, \dot{\Sigma}_0)$ *is given by:*

$$\exp_{\mathcal{N}(0,I)}(s\dot{\mu}_0, s\dot{\Sigma}_0) = \mathcal{N}\left(2R(s)\mathrm{sh}(\frac{sG}{2})G^- \dot{\mu}_0, R(s)R(s)^T\right), \quad (43)$$

where exp *is the Riemannian exponential of* $\mathbb{G}_d$*,* G *is any matrix satisfying:*

$$G^2 = \dot{\Sigma}_0^2 + 2\dot{\mu}_0\dot{\mu}_0^T, \tag{44}$$

$$R(s) = \left(\left(\mathrm{ch}(\frac{sG}{2}) - \dot{\Sigma}_0 G^- \mathrm{sh}(\frac{sG}{2})\right)^{-1}\right)^T \tag{45}$$

and G^- *is a pseudo-inverse of* G

In [5], the existence of G (as a square root of $\dot{\Sigma}_0^2 + 2\dot{\mu}_0\dot{\mu}_0^T$) is proven. Notice that, anyway, in the expansions of (43) and (45), only even powers of G appear.

Additionally, since, for all $A \in GL_d(\mathbb{R})$, for all $\mu_0 \in \mathbb{R}^d$, the application:

$$\begin{array}{cccc} \phi & : & \mathbb{G}_d & \rightarrow & \mathbb{G}_d \\ & & \mathcal{N}(\mu, \Sigma) & \mapsto & \mathcal{N}(A\mu + \mu_0, A\Sigma A^T) \end{array} \tag{46}$$

preserves the geodesics, we find the general expression for the geodesics of $\mathbb{G}_d$.

Corollary 1. *Let* $\mu_0 \in \mathbb{R}^d$*,* $A \in GL_d(\mathbb{R})$ *and* $(\dot{\mu}_0, \dot{\Sigma}_0) \in T_{\mathcal{N}(\mu_0, A_0A_0^T)}\mathbb{G}_d$*. The geodesic of* $\mathbb{G}_d$ *starting from* $\mathcal{N}(\mu, \Sigma)$ *with initial speed* $(\dot{\mu}_0, \dot{\Sigma}_0)$ *is given by:*

$$\exp_{\mathcal{N}(\mu_0, A_0A_0^T)}(s\dot{\mu}_0, s\dot{\Sigma}_0) = \mathcal{N}(\mu_1, A_1A_1^T), \tag{47}$$

with:

$$\mu_1 = 2A_0R(s)\mathrm{sh}(\frac{sG}{2})G^- A_0^{-1}\dot{\mu}_0 + \mu_0, \tag{48}$$

$$A_1 = A_0R(s), \tag{49}$$

where exp *is the Riemannian exponential of* $\mathbb{G}_d$*,* G *is any matrix satisfying:*

$$G^2 = A_0^{-1}(\dot{\Sigma}_0\Sigma_0^{-1}\dot{\Sigma}_0 + 2\dot{\mu}_0\dot{\mu}_0^T)(A_0^{-1})^T, \tag{50}$$

$$R(s) = \left(\left(\mathrm{ch}(\frac{sG}{2}) - A_0^{-1}\dot{\Sigma}_0(A_0^{-1})^T G^- \mathrm{sh}(\frac{sG}{2})\right)^{-1}\right)^T, \tag{51}$$

and G^- *is a pseudo-inverse of* G*.*

It should be noted that the final values for mean and covariance do not depend on the choice of G as a square root of:

$$A_0^{-1}(\dot{\Sigma}_0\Sigma_0^{-1}\dot{\Sigma}_0 + 2\dot{\mu}_0\dot{\mu}_0^T)(A_0^{-1})^T.$$

The reason for this is that $\mathrm{ch}(G)$ is a Taylor series in G^2, and so are $\mathrm{sh}(G)G^-$ and $G^-\mathrm{sh}(G)$.

For our practical implementation, we actually used these Taylor series instead of the expression of the corollary.

6. Comparing GIGO, xNES and Pure Rank-μ CMA-ES

6.1. Definitions

In this section, we recall the xNES and pure rank-μ CMA-ES, and we describe them in the IGO framework, thus allowing a reasonable comparison with the GIGO algorithms.

6.1.1. xNES

We recall a restriction of the xNES algorithm, introduced in [19] (this restriction is sufficient to describe the numerical experiments in [19]).

Definition 9 (xNES algorithm)**.** *The xNES algorithm with sample size N, weights w_i and learning rates η_μ and η_Σ updates the parameters $\mu \in \mathbb{R}^d$, $A \in M_d(\mathbb{R})$ with the following rule: At each step, N points $x_1, ..., x_N$ are sampled from the distribution $\mathcal{N}(\mu, AA^T)$. Without loss of generality, we assume $f(x_1) < ... < f(x_N)$. The parameter is updated according to:*

$$\mu \leftarrow \mu + \eta_\mu A G_\mu,$$

$$A \leftarrow A \exp(\eta_\Sigma G_M / 2),$$

where, setting $z_i = A^{-1}(x_i - \mu)$:

$$G_\mu = \sum_{i=1}^{N} w_i z_i,$$

$$G_M = \sum_{i=1}^{N} w_i (z_i z_i^T - I).$$

The more general version decomposes the matrix A as σB, where $\det B = 1$, and uses two different learning rates for σ and for B. We gave the version where these two learning rates are equal (in particular, for the default parameters in [19], these two learning rates are equal). This restriction of the xNES algorithm can be described in the IGO framework, provided all of the learning rates are equal (most of the elements of the proof can be found in [19] (the proposition below essentially states that xNES is a natural gradient update) or in [1]):

Proposition 6 (xNES as IGO)**.** *The xNES algorithm with sample size N, weights w_i and learning rates $\eta_\mu = \eta_\Sigma = \delta t$ coincides with the IGO algorithm with sample size N, weights w_i, step size δt and in which, given the current position (μ_t, A_t), the set of Gaussians is parametrized by:*

$$\phi_{\mu_t, A_t} : (\delta, M) \mapsto \mathcal{N}\left(\mu_t + A_t\delta, \left(A_t \exp(\frac{1}{2}M)\right)\left(A_t \exp(\frac{1}{2}M)\right)^T\right),$$

with $\delta \in \mathbb{R}^m$ and $M \in \mathrm{Sym}(\mathbb{R}^m)$.

The parameters maintained by the algorithm are (μ, A), and the x_i are sampled from $\mathcal{N}(\mu, AA^T)$.

Proof. Let us compute the IGO update in the parametrization ϕ_{μ_t,A_t}: we have $\delta^t = 0$, $M^t = 0$, and by using Proposition 1, we can see that for this parametrization, the Fisher information matrix at $(0,0)$ is the identity matrix. The IGO update is therefore,

$$(\delta, M)^{t+\delta t} = (\delta, M)^t + \delta t Y_\delta(\delta, M) + \delta t Y_M(\delta, M) = \delta t Y_\delta(\delta, M) + \delta t Y_M(\delta, M),$$

where:

$$Y_\delta(\delta, M) = \sum_{i=1}^{N} w_i \nabla_\delta \ln(p(x_i|(\delta, M))$$

and:

$$Y_M(\delta, M) = \sum_{i=1}^{N} w_i \nabla_M \ln(p(x_i|(\delta, M)).$$

Since $\operatorname{tr}(M) = \log(\det(\exp(M)))$, we have:

$$\ln P_{\delta,M}(x) = -\frac{d}{2}\ln(2\pi) - \ln(\det A) - \frac{1}{2}\operatorname{tr} M - \frac{1}{2}\|\exp(-\frac{1}{2}M)A^{-1}(x - \mu - A\delta)\|^2,$$

and a straightforward computation yields:

$$Y_\delta(\delta, M) = \sum_{i=1}^{N} w_i z_i = G_\mu,$$

and:

$$Y_M(\delta, M) = \frac{1}{2}\sum_{i=1}^{N} w_i(z_i z_i^T - I) = G_M.$$

Therefore, the IGO update is:

$$\delta(t + \delta t) = \delta(t) + \delta t G_\mu,$$

$$M(t + \delta t) = M(t) + \delta t G_M,$$

or, in terms of mean and covariance matrix:

$$\mu(t + \delta t) = \mu(t) + \delta t A(t) G_\mu$$

$$A(t + \delta t) = A(t)\exp(\delta t G_M/2),$$

or:

$$\Sigma(t + \delta t) = A(t)\exp(\delta t G_M)A(t)^T.$$

This is the xNES update. □

6.1.2. Using a Square Root of the Covariance Matrix

Firstly, we recall that the IGO framework (on $\mathbb{G}_d$, for example) emphasizes the Riemannian manifold structure on $\mathbb{G}_d$. All of the algorithms studied here (including GIGO, which is not strictly speaking an IGO algorithm) define a trajectory in $\mathbb{G}_d$ (a new point for each step), and to go from a point θ to the next one (θ'), we follow some curve $\gamma : [0, \delta t] \to \mathbb{G}_d$, with $\gamma(0) = \theta$, $\gamma(\delta t) = \theta'$ and $\dot{\gamma}(0)$ given by the natural gradient ($\dot{\gamma}(0) = \sum_{i=1}^{N} \hat{w}_i \tilde{\nabla}_\theta P_\theta(x_i) \in T_\theta \mathbb{G}_d$).

To be compatible with this point of view, an algorithm giving an update rule for a square root (any matrix A such that $\Sigma = AA^T$: since we do not force A to be symmetric, the decomposition is not unique) of the covariance matrix A has to satisfy the following condition: for a given initial speed, the covariance matrix $\Sigma^{t+\delta t}$ after one step must depend only on Σ^t and not on the square root A^t chosen for Σ^t.

The xNES algorithm does satisfy this condition: consider two xNES algorithms, with the same learning rates, respectively, at (μ, A_1^t) and (μ, A_2^t), with $A_1^t(A_1^t)^T = A_2^t(A_2^t)^T$ (*i.e.*, they define the same Σ^t), using the same samples x_i to compute the natural gradient update , then we will have $\Sigma_1^{t+\delta t} = \Sigma_2^{t+\delta t}$. Using the definitions of Section 6.3, we have just shown that what we will call the "xNES trajectory" is well defined.

It is also important to notice that, in order to be well defined, a natural gradient algorithm updating a square root of the covariance matrix has to specify more conditions than simply following the natural gradient.

The reason for this is that the natural gradient is a vector tangent to $\mathbb{G}_d$: it lives in a space of dimension $d(d+3)/2$ (the dimension of $\mathbb{G}_d$), whereas the vector (μ, A) lives in a space of dimension $d(d+1)$ (the dimension of $\mathbb{R}^n \times GL_n(\mathbb{R})$), which is too large: there exists infinitely many applications $t \mapsto A_t$, such that a given curve $\gamma : t \mapsto \mathcal{N}(\mu_t, \Sigma_t)$ can be written $\gamma(t) = \mathcal{N}(\mu_t, A_t A_t^T)$. This is why Theorem 5 is simply an implication, whereas Theorem 4 is an equivalence.

More precisely, let us consider A in $\mathrm{GL}_d(\mathbb{R})$ and v_A, v'_A two infinitesimal updates of A. Since $\Sigma = AA^T$, the infinitesimal update of Σ corresponding to v_A (resp. v'_A) is $v_\Sigma = Av_A^T + v_A A^T$ (resp. $v'_\Sigma = Av_A'^T + v'_A A^T$).

It is now easy to see that v_A and v'_A define the same direction for Σ (*i.e.*, $v_\Sigma = v'_\Sigma$) if and only if $AM^T + MA^T = 0$, where $M = v_A - v'_A$. This is equivalent to $A^{-1}M$ antisymmetric.

For any $A \in \mathrm{M}_d(\mathbb{R})$, let us denote by T_A the space of the matrices M, such that $A^{-1}M$ is antisymmetric or, in other words, $T_A := \{u \in \mathrm{M}_d(\mathbb{R}),\ Au^T + uA^T = 0\}$. Having a subspace S_A in direct sum with T_A for all A is sufficient (but not necessary) to have a well-defined update rule. Namely, consider the (linear) application:

$$\begin{array}{cccc} \phi_A & : & \mathrm{M}_d(\mathbb{R}) & \to & \mathrm{S}_d(\mathbb{R}) \\ & & v_A & \mapsto & Av_A^T + v_A A^T \end{array},$$

sending an infinitesimal update of A to the corresponding update of Σ. It is not bijective, but as we have seen before, $\mathrm{Ker}\, \phi_A = T_A$, and therefore, if we have, for some U_A,

$$\mathrm{M}_d(\mathbb{R}) = U_A \oplus T_A, \tag{52}$$

then $\phi_A|_{U_A}$ is an isomorphism. Let v_Σ be an infinitesimal update of Σ. We choose the following update of A corresponding to v_Σ:

$$v_A := (\phi_A|_{U_A})^{-1}(v_\Sigma). \tag{53}$$

Any U_A, such that $U_A \oplus T_A = \mathrm{M}_d(\mathbb{R})$, is a reasonable choice to pick v_A for a given v_Σ. The choice $S_A = \{u \in \mathrm{M}_d(\mathbb{R}),\ Au^T - uA^T = 0\}$ has an interesting additional property; it is the orthogonal of T_A for the norm:

$$\|v_A\|_\Sigma^2 := \mathrm{Tr}(v_A^T \Sigma^{-1} v_A) = \mathrm{Tr}((A^{-1}v_A)^T A^{-1} v_A). \tag{54}$$

and consequently, it can be defined without referring to the parametrization, which makes it a canonical choice. To prove this, remark that $T_A = \{M \in \mathrm{M}_d(\mathbb{R}), A^{-1}M$ antisymmetric$\}$ and $S_A = \{M \in \mathrm{M}_d(\mathbb{R}), A^{-1}M$ symmetric$\}$ and that if M is symmetric and N is antisymmetric, then

$$\mathrm{Tr}(M^T N) = \sum_{i,j=1}^{d} m_{ij}n_{ij} = \sum_{i=1}^{d} m_{ii}n_{ii} + \sum_{1 \leqslant i < j \leqslant d} m_{ij}(n_{ij} + n_{ji}) = 0. \tag{55}$$

Let us now show that this is the choice made by xNES and GIGO-A (which are well-defined algorithms updating a square root of the covariance matrix).

Proposition 7. *Let $A \in M_n(\mathbb{R})$. The v_A given by the xNES and GIGO-A algorithms lies in $S_A = \{u \in$ $\mathrm{M}_d(\mathbb{R})$, $Au^T - uA^T = 0\} = S_A$.*

Proof. For xNES, let us write $\dot\gamma(0) = (v_\mu, v_\Sigma)$ and $v_A := \frac{1}{2}AG_M$. We have $A^{-1}v_A = \frac{1}{2}G_M$, and therefore, forcing M (and G_M) to be symmetric in xNES is equivalent to $A^{-1}v_A = (A^{-1}v_A)^T$, which can be rewritten as $Av_A^T = v_A A^T$. For GIGO-A, Equation (40) shows that $\Sigma_t(J_\Sigma - J_\mu \mu_t^T)$ is symmetric, and with this fact in mind, Equation (42) shows that we have $Av_A^T = v_A A^T$ (v_A is $\dot A_t$). □

6.1.3. Pure Rank-μ CMA-ES

We now recall the pure rank-μ CMA-ES algorithm. The general CMA-ES algorithm is described in [21].

Definition 10 (Pure rank-μ CMA-ES algorithm)**.** *The pure rank-μ CMA-ES algorithm with sample size N, weights w_i and learning rates η_μ and η_Σ is defined by the following update rule: At each step, N points $x_1, ..., x_N$ are sampled from the distribution $\mathcal{N}(\mu, \Sigma)$. Without loss of generality, we assume $f(x_1) < ... < f(x_N)$. The parameter is updated according to:*

$$\mu \leftarrow \mu + \eta_\mu \sum_{i=1}^{N} w_i(x_i - \mu),$$

$$\Sigma \leftarrow \Sigma + \eta_\Sigma \sum_{i=1}^{N} w_i((x_i - \mu)(x_i - \mu)^T - \Sigma).$$

The pure rank-μ CMA-ES can also be described in the IGO framework; see, for example, [20].

Proposition 8 (Pure rank-μ CMA-ES as IGO)**.** *The pure rank-μ CMA-ES algorithm with sample size N, weights w_i and learning rates $\eta_\mu = \eta_\Sigma = \delta t$ coincides with the IGO algorithm with sample size N, weights w_i, step size δt and the parametrization (μ, Σ).*

6.2. Twisting the Metric

As we can see, the IGO framework does not allow one to recover the learning rates for xNES and pure rank-μ CMA-ES, which is a problem, since usually, the covariance learning rate is set much smaller than the mean learning rate (see either [19] or [21]).

A way to recover these learning rates is to incorporate them directly into the metric (see also blockwise GIGO, in Section 6.4). More precisely:

Definition 11 (Twisted Fisher metric)**.** *Let $\eta_\mu, \eta_\Sigma \in \mathbb{R}$, and let $(P_\theta)_{\theta\in\Theta}$ be a family of normal probability distributions: $P_\theta = \mathcal{N}(\mu(\theta), \Sigma(\theta))$, with μ and Σ C^1. We call the "(η_μ, η_Σ)-twisted Fisher metric" the metric defined by:*

$$I_{i,j}(\eta_\mu, \eta_\Sigma)(\theta) = \frac{1}{\eta_\mu}\frac{\partial \mu^T}{\partial \theta_i}\Sigma^{-1}\frac{\partial \mu}{\partial \theta_j} + \frac{1}{\eta_\Sigma}\frac{1}{2}\operatorname{tr}\left(\Sigma^{-1}\frac{\partial \Sigma}{\partial \theta_i}\Sigma^{-1}\frac{\partial \Sigma}{\partial \theta_j}\right). \tag{56}$$

All of the remainder of this section is simply a rewriting of the work in Section 2 with the twisted Fisher metric instead of the regular Fisher metric. We will use the term "twisted geodesic" instead of "geodesic for the twisted metric".

This approach seems to be somewhat arbitrary: arguably, the mean and the covariance play a "different role" in the definition of a Gaussian (only the covariance can affect diversity, for example), but we lack a reasonable intrinsic characterization that would make this choice of twisting more natural. This construction can be slightly generalized (see the Appendix).

The IGO flow and the IGO algorithms can be modified to take into account the twisting of the metric; the (η_μ, η_Σ)-twisted IGO flow reads:

$$\frac{\mathrm{d}\theta^t}{\mathrm{d}t} = I(\eta_\mu, \eta_\Sigma)^{-1}(\theta) \int_X W^f_{\theta^t}(x)\nabla_\theta \ln P_\theta(x) P_{\theta^t}(\mathrm{d}x). \tag{57}$$

The only difference with (9) is that $I^{-1}(\theta)$ has been replaced by $I(\eta_\mu, \eta_\Sigma)^{-1}(\theta)$.

This leads us to the twisted IGO algorithms.

Definition 12. *The (η_μ, η_Σ)-twisted IGO algorithm associated with parametrization θ, sample size N, step size δt and selection scheme w is given by the following update rule:*

$$\theta^{t+\delta t} = \theta^t + \delta t I(\eta_\mu, \eta_\Sigma)^{-1}(\theta^t) \sum_{i=1}^{N} \hat{w}_i \frac{\partial \ln P_\theta(x_i)}{\partial \theta}.$$

Definition 13. *The (η_μ, η_Σ)-twisted geodesic IGO algorithm associated with sample size N, step size δt and selection scheme w is given by the following update rule:*

$$\theta^{t+\delta t} = \exp_{\theta^t}(Y\delta t) \tag{58}$$

where:

$$Y = I(\eta_\mu, \eta_\Sigma)^{-1}(\theta^t) \sum_{i=1}^{N} \hat{w}_i \frac{\partial \ln P_\theta(x_i)}{\partial \theta}. \tag{59}$$

By definition, the twisted geodesic IGO algorithm does not depend on the parametrization (but it does depend on η_μ and η_Σ).

There is some redundancy between δt, η_μ and η_Σ: the only values actually appearing in the equations are $\delta t \eta_\mu$ and $\delta t \eta_\Sigma$. More formally:

Proposition 9. *Let $k, d, N \in \mathbb{N}$, $\eta_\mu, \eta_\Sigma, \delta t, \lambda_1, \lambda_2 \in \mathbb{R}$ and $w\colon [0;1] \to \mathbb{R}$.*

The (η_μ, η_Σ)-twisted IGO algorithm with sample size N, step size δt and selection scheme w coincides with the $(\lambda_1\eta_\mu, \lambda_1\eta_\Sigma)$-twisted IGO algorithm with sample size N, step size $\lambda_2\delta t$ and selection scheme $\frac{1}{\lambda_1\lambda_2}w$. The same is true for geodesic IGO.

In order to obtain the twisted algorithms, the Fisher metric in IGO has to be replaced by the metric from Definition 11. In practice, the equations found by twisting the metric are exactly the equations without twisting, except that we have "forced" the learning rates η_μ, η_Σ to appear by multiplying the increments of μ and Σ by η_μ and η_Σ.

We can now describe pure rank-μ CMA-ES and xNES with separate learning rates as twisted IGO algorithms:

Proposition 10 (xNES as IGO)**.** *The xNES algorithm with sample size N, weights w_i and learning rates $\eta_\mu, \eta_\sigma = \eta_B = \eta_\Sigma$ coincides with the $\frac{\eta_\mu}{\delta t}, \frac{\eta_\Sigma}{\delta t}$-twisted IGO algorithm with sample size N, weights w_i, step size δt and in which, given the current position (μ_t, A_t), the set of Gaussians is parametrized by:*

$$(\delta, M) \mapsto \mathcal{N}\left(\mu_t + A_t\delta, \left(A_t \exp(\frac{1}{2}M)\right)\left(A_t \exp(\frac{1}{2}M)\right)^T\right),$$

with $\delta \in \mathbb{R}^m$ and $M \in \mathrm{Sym}(\mathbb{R}^m)$.

The parameters maintained by the algorithm are (μ, A), and the x_i are sampled from $\mathcal{N}(\mu, AA^T)$.

Proposition 11 (Pure rank-μ CMA-ES as IGO)**.** *The pure rank-μ CMA-ES algorithm with sample size N, weights w_i and learning rates η_μ and η_Σ coincides with the $(\frac{\eta_\mu}{\delta t}, \frac{\eta_\Sigma}{\delta t})$-twisted IGO algorithm with sample size N, weights w_i, step size δt and the parametrization (μ, Σ).*

The proofs of these two statements are an easy rewriting of their non-twisted counterparts: one can return to the non-twisted metric (up to a η_Σ factor) by changing μ to $\frac{\sqrt{\eta_\sigma}}{\sqrt{\eta_\mu}}\mu$.

We give the equations of the twisted geodesics of $\mathbb{G}_d$ in the Appendix.

6.3. Trajectories of Different IGO Steps

As we have seen, two different IGO algorithms (or an IGO algorithm and the GIGO algorithm) coincide at first order in δt when $\delta t \to 0$. In this section, we study the differences between pure rank-μ CMA-ES, xNES and GIGO by looking at the second order in δt, and in particular, we show that xNES and GIGO do not coincide in the general case.

We view the updates done by one step of the algorithms as paths on the manifold $\mathbb{G}_d$, from $(\mu(t), \Sigma(t))$ to $(\mu(t + \delta t), \Sigma(t + \delta t))$, where δt is the time step of our algorithms, seen as IGO algorithms. More formally:

Definition 14. *(1) We call the GIGO update trajectory the application:*

$$T_{\mathrm{GIGO}}\colon (\mu, \Sigma, v_\mu, v_\Sigma) \mapsto \left(\delta t \mapsto \exp_{\mathcal{N}(\mu, AA^T)}(\delta t \eta_\mu v_\mu, \delta t \eta_\Sigma v_\Sigma)\right).$$

*(*exp *is the exponential of the Riemannian manifold $\mathbb{G}_d(\eta_\mu, \eta_\Sigma)$)*

(2) We call the xNES update trajectory the application:

$$T_{\mathrm{xNES}}\colon (\mu, \Sigma, v_\mu, v_\Sigma) \mapsto \left(\delta t \mapsto \mathcal{N}(\mu + \delta t \eta_\mu v_\mu, A\exp[\eta_\Sigma \delta t A^{-1} v_\Sigma (A^{-1})^T]A^T)\right),$$

with $AA^T = \Sigma$. The application above does not depend on the choice of a square root A.

(3) We call the CMA-ES update trajectory the application:

$$T_{\mathrm{CMA}}\colon (\mu, \Sigma, v_\mu, v_\Sigma) \mapsto \left(\delta t \mapsto \mathcal{N}(\mu + \delta t \eta_\mu v_\mu, AA^T + \delta t \eta_\Sigma v_\Sigma)\right).$$

These applications map the set of tangent vectors to $\mathbb{G}_d$ ($T\mathbb{G}_d$) to the curves in $\mathbb{G}_d(\eta_\mu, \eta_\Sigma)$.

We will also use the following notation: $\mu_{\mathrm{GIGO}} := \phi_\mu \circ T_{GIGO}$, $\mu_{\mathrm{xNES}} := \phi_\mu \circ T_{xNES}$, $\mu_{\mathrm{CMA}} := \phi_\mu \circ T_{CMA}$, $\Sigma_{\mathrm{GIGO}} := \phi_\Sigma \circ T_{GIGO}$, $\Sigma_{\mathrm{xNES}} := \phi_\Sigma \circ T_{xNES}$ and $\Sigma_{\mathrm{CMA}} := \phi_\Sigma \circ T_{CMA}$, where ϕ_μ (resp. ϕ_Σ) extracts the μ-component (resp. the Σ-component) of a curve.

In particular, $\mathrm{Im}(\phi_\mu) \subset \mathbb{R}^d$ and $\mathrm{Im}(\phi_\Sigma) \subset P_d$, where P_d (the set of real symmetric positive-definitematrices of dimension d) is seen as a subset of $\mathbb{R}^{d^2}$.

For instance, $T_{\mathrm{GIGO}}(\mu, \Sigma, v_\mu, v_\Sigma)(\delta t)$ gives the position (mean and covariance matrix) of the GIGO algorithm after a step of size δt, while μ_{GIGO} and Σ_{GIGO} give, respectively, the mean component and the covariance component of this position.

This formulation ensures that the trajectories we are comparing had the same initial position and the same initial speed, which is the case provided the sampled points (the values directly sampled from $\mathcal{N}(\mu, \Sigma)$, not from $\mathcal{N}(0, I)$ and transformed) are the same.

Different IGO algorithms coincide at first order in δt. The following proposition gives the second order expansion of the trajectories of the algorithms.

Proposition 12 (Second derivatives of the trajectories)**.** *We have:*

$$\mu_{\mathrm{GIGO}}(\mu, \Sigma, v_\mu, v_\Sigma)''(0) = \eta_\mu \eta_\Sigma v_\Sigma \Sigma_0^{-1} v_\mu,$$

$$\mu_{\mathrm{xNES}}(\mu, \Sigma, v_\mu, v_\Sigma)''(0) = \mu_{\mathrm{CMA}}(\mu, \Sigma, v_\mu, v_\Sigma)''(0) = 0,$$

$$\Sigma_{\mathrm{GIGO}}(\mu, \Sigma, v_\mu, v_\Sigma)''(0) = \eta_\Sigma^2 v_\Sigma \Sigma^{-1} v_\Sigma - \eta_\mu \eta_\Sigma v_\mu v_\mu^T,$$

$$\Sigma_{\mathrm{xNES}}(\mu, \Sigma, v_\mu, v_\Sigma)''(0) = \eta_\Sigma^2 v_\Sigma \Sigma^{-1} v_\Sigma,$$

$$\Sigma_{\mathrm{CMA}}(\mu, \Sigma, v_\mu, v_\Sigma)''(0) = 0.$$

Proof. We can immediately see that the second derivatives of μ_{xNES}, μ_{CMA} and Σ_{CMA} are zero. Next, we have:

$$\begin{aligned}\Sigma_{\mathrm{xNES}}(\mu, \Sigma, v_\mu, v_\Sigma)(t) &= A\exp[tA^{-1}\eta_\Sigma v_\Sigma (A^{-1})^T]A^T \\ &= AA^T + t\eta_\Sigma v_\Sigma + \frac{t^2}{2}\eta_\Sigma^2 v_\Sigma (A^{-1})^T A^{-1} v_\Sigma + o(t^2) \\ &= \Sigma + t\eta_\Sigma v_\Sigma + \frac{t^2}{2}\eta_\Sigma^2 v_\Sigma \Sigma^{-1} v_\Sigma + o(t^2).\end{aligned}$$

The expression of $\Sigma_{\mathrm{xNES}}(\mu, \Sigma, v_\mu, v_\Sigma)''(0)$ follows.

Now, for GIGO, let us consider the geodesic starting at (μ_0, Σ_0) with initial speed $(\eta_\mu v_\mu, \eta_\Sigma v_\Sigma)$. By writing $J_\mu(0) = J_\mu(t)$, we find $\dot{\mu}(t) = \Sigma(t)\Sigma_0^{-1}\dot{\mu}_0$. We then easily have $\ddot{\mu}(0) = \dot{\Sigma}_0\Sigma_0^{-1}\dot{\mu}_0$. In other words:

$$\mu_{\mathrm{GIGO}}(\mu, \Sigma, v_\mu, v_\Sigma)''(0) = \eta_\mu \eta_\Sigma v_\Sigma \Sigma_0^{-1} v_\mu.$$

Finally, by using Theorem 4 and differentiating, we find:

$$\ddot{\Sigma} = \eta_\Sigma \dot{\Sigma}(J_\Sigma - J_\mu \mu^T) - \eta_\Sigma \Sigma J_\mu \dot{\mu}^T,$$

$$\ddot{\Sigma}_0 = \eta_\Sigma \dot{\Sigma}_0 \frac{1}{\eta_\Sigma} \Sigma_0^{-1} \dot{\Sigma}_0 - \frac{\eta_\Sigma}{\eta_\mu} \dot{\mu}_0 \dot{\mu}_0^T = \eta_\Sigma^2 v_\Sigma \Sigma_0^{-1} v_\Sigma - \eta_\Sigma \eta_\mu v_\mu v_\mu^T.$$

□

In order to interpret these results, we will look at what happens in dimension one. In higher dimensions, we can suppose that the algorithms exhibit a similar behavior, but an exact interpretation is more difficult for GIGO in $\mathbb{G}_d$.

- In [19], it has been noted that xNES converges to quadratic minima slower than CMA-ES and that it is less subject to premature convergence. That fact can be explained by observing that the mean update is exactly the same for CMA-ES and xNES, whereas xNES tends to have a higher variance (Proposition 12 shows this at order two, and it is easy to see that in dimension one, for any μ, Σ, v_μ, v_Σ, we have $\Sigma_{\mathrm{xNES}}(\mu, \Sigma, v_\mu, v_\Sigma) > \Sigma_{\mathrm{CMA}}(\mu, \Sigma, v_\mu, v_\Sigma)$).
- At order two, GIGO moves the mean faster than xNES and CMA-ES if the standard deviation is increasing and more slowly if it is decreasing. This seems to be a reasonable behavior (if the covariance is decreasing, then the algorithm is presumably close to a minimum, and it should not leave the area too quickly). This remark holds only for isolated steps, because we do not take into account the evolution of the variance.
- The geodesics of $\mathbb{G}_1$ are half-circles (see Figure 2 below; we recall that $\mathbb{G}_1$ is the Poincaré half-plane). Consequently, if the mean is supposed to move (which always happens), then $\sigma \to 0$ when $\delta t \to \infty$. For example, a step whose initial speed has no component on the standard deviation will always decrease it. See also Proposition 15, about the optimization of a linear function.
- For the same reason, for a given initial speed, the update of μ always stays bounded as a function of δt: it is not possible to make one step of the GIGO algorithm go further than a fixed point by increasing δt. Still, the geodesic followed by GIGO changes at each step, so the mean of the overall algorithm is not bounded.

We now show that xNES follows the geodesics of $\mathbb{G}_d$ if the mean is fixed, but that xNES and GIGO do not coincide otherwise.

Proposition 13 (xNES is not GIGO in the general case)**.** *Let* $\mu, v_\mu \in \mathbb{R}^d$, $A \in \mathrm{GL}_d$, $v_\Sigma \in \mathrm{M}_d$.

Then, the GIGO and xNES updates starting at $\mathcal{N}(\mu, \Sigma)$ *with initial speeds* v_μ *and* v_Σ *follow the same trajectory if and only if the mean remains constant. In other words:*

$T_{\mathrm{GIGO}}(\mu, \Sigma, v_\mu, v_\Sigma) = T_{\mathrm{xNES}}(\mu, \Sigma, v_\mu, v_\Sigma)$ *if and only if* $v_\mu = 0$.

Proof. If $v_\mu = 0$, then we can compute the GIGO update by using Theorem 4: since $J_\mu = 0$, $\dot{\mu} = 0$, and μ remains constant. Now, we have $J_\Sigma = \Sigma^{-1}\dot{\Sigma}$; this is enough information to compute the update. Since

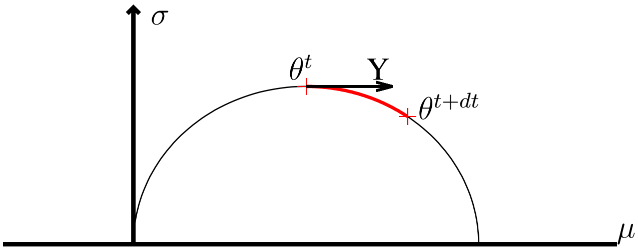

Figure 2. One step of the geodesic IGO (GIGO) update.

this quantity is also preserved by the xNES algorithm (see, for example, the proof of Proposition 14), the two updates coincide.

If $v_\mu \neq 0$, then $\Sigma_{\text{xNES}}(\mu, \Sigma, v_\mu, v_\Sigma)''(0) - \Sigma_{\text{GIGO}}(\mu, \Sigma, v_\mu, v_\Sigma)''(0) = \eta_\mu \eta_\Sigma v_\mu v_\mu^T \neq 0$ and, in particular, $T_{\text{GIGO}}(\mu, \Sigma, v_\mu, v_\Sigma) \neq T_{\text{xNES}}(\mu, \Sigma, v_\mu, v_\Sigma)$. □

6.4. Blockwise GIGO

Although xNES is not GIGO, it is possible to define a family of algorithms extending GIGO and including xNES, by decomposing our family of probability distributions as a product and by following the restricted geodesics simultaneously.

Definition 15 (Splitting)**.** *Let Θ be a Riemannian manifold. A splitting of Θ is n manifolds $\Theta_1, ..., \Theta_n$ and a diffeomorphism $\Theta \cong \Theta_1 \times ... \times \Theta_n$. If for all $x \in \Theta$, for all $1 \leqslant i < j \leqslant n$, we also have $T_{i,x}M \perp T_{j,x}M$ as subspaces of T_xM (see Notation 2), then the splitting is said to be compatible with the Riemannian structure. If the Riemannian manifold is not ambiguous, we will simply write a "compatible splitting".*

We now give some notation, and we define the blockwise GIGO update:

Notation 2. *Let Θ be a Riemannian manifold, $\Theta_1, ..., \Theta_n$ a splitting of Θ, $\theta = (\theta_1, ..., \theta_n) \in \Theta$, $Y \in T_\theta\Theta$ and $1 \leqslant i \leqslant n$.*

- *We denote by $\Theta_{\theta,i}$ the Riemannian manifold*

$$\{\theta_1\} \times ... \times \{\theta_{i-1}\} \times \Theta_i \times \{\theta_{i+1}\} \times ... \times \{\theta_n\},$$

with the metric induced from Θ. There is a canonical isomorphism of vector spaces $T_\theta\Theta = \oplus_{i=1}^n T\Theta_{\theta,i}$. Moreover, if the splitting is compatible, it is an isomorphism of Euclidean spaces.

- *We denote by $\Phi_{\theta,i}$ the exponential at θ of the manifold $\Theta_{\theta,i}$.*

Definition 16 (Blockwise GIGO update)**.** *Let $\Theta_1, ..., \Theta_n$ be a compatible splitting. The blockwise GIGO algorithm in Θ with splitting $\Theta_1, ..., \Theta_n$ associated with sample size N, step sizes $\delta t_1, ..., \delta t_n$ and selection scheme w is given by the following update rule:*

$$\theta \leftarrow (\theta_1^{t+\delta t_1}, ..., \theta_n^{t+\delta t_n}) \quad (60)$$

where:

$$Y = I^{-1}(\theta^t) \sum_{i=1}^{N} \hat{w}_i \frac{\partial \ln P_\theta(x_i)}{\partial \theta}, \tag{61}$$

$$\theta_k^{t+\delta t_k} = \Phi_{\theta^t,k}(\delta t_k Y_k), \tag{62}$$

with Y_k the $T\Theta_{\theta,k}$-component of Y. This update only depends on the splitting (and not on the parametrization inside each Θ_k).

The compatibility condition ensures that the natural gradient of $W_{\theta^t}^f$ (defined in Section 2.2) in the whole manifold Θ really is the sum of the gradients of this same function in the submanifolds Θ_k. A practical consequence is that the Y_k in Equation (62) can be computed simply by taking the natural gradient in Θ_k:

$$Y_k = I_k^{-1}(\theta_i^t) \sum_{i=1}^{N} \hat{w}_i \frac{\partial \ln P_\theta(x_i)}{\partial \theta_k}, \tag{63}$$

where I_k is the metric of Θ_k.

Since blockwise GIGO only depends on the splitting (and the tunable parameters: sample size, step sizes and selection scheme), it can be thought of as almost parametrization-invariant.

Notice that blockwise GIGO updates and twisted GIGO updates are two different things: firstly, blockwise GIGO can be defined on any manifold with a compatible splitting, whereas twisted GIGO (and twisted IGO) are only defined for Gaussians. However, even in $\mathbb{G}_d(\eta_\mu, \eta_\Sigma)$, with the splitting (μ, Σ), these two algorithms are different: for instance, if $\eta_\mu = \eta_\Sigma$ and $\delta t = 1$, then the twisted GIGO is the regular GIGO algorithm, whereas blockwise GIGO is not (actually, we will prove that it is the xNES algorithm). The only thing blockwise GIGO and twisted GIGO have in common is that they are compatible with the (η_μ, η_Σ)-twisted IGO flow Equation (57): a parameter θ^t following these updates with $\delta t \to 0$ and $N \to \infty$ is a solution of Equation (57).

We now have a new description of the xNES algorithm:

Proposition 14 (xNES is a Blockwise GIGO algorithm)**.** *The Blockwise GIGO algorithm in $\mathbb{G}_d$ with splitting $\Phi : \mathcal{N}(\mu, \Sigma) \mapsto (\mu, \Sigma)$, sample size N, step sizes $\delta t_\mu, \delta t_\Sigma$ and selection scheme w coincides with the xNES algorithm with sample size N, weights w_i and learning rates $\eta_\mu = \delta t_\mu, \eta_\sigma = \eta_B = \delta t_\Sigma$.*

Proof. Firstly, notice that the splitting (μ, Σ) is compatible, by Proposition 1.

Now, let us compute the Blockwise GIGO update: we have $\mathbb{G}_d \cong \mathbb{R}^d \times P_d$, where P_d is the space of real positive-definite matrices of dimension d. We have $\Theta_{\theta^t,1} = (\mathbb{R}^d \times \{\Sigma^t\}) \hookrightarrow \mathbb{G}_d$, $\Theta_{\theta^t,2} = (\{\mu^t\} \times P_d) \hookrightarrow \mathbb{G}_d$. The induced metric on $\Theta_{\theta^t,1}$ is the Euclidean metric, so we have:

$$\mu \leftarrow \mu^t + \delta t_1 Y_\mu.$$

Since we have already shown (using the notation in Definition 9) that $Y_\mu = AG_\mu$ (in the proof of Proposition 6), we find:

$$\mu \leftarrow \mu^t + \delta t_1 AG_\mu.$$

On $\Theta_{\theta^t,2}$, we have the following Lagrangian for the geodesics:

$$\mathcal{L}(\Sigma, \dot{\Sigma}) = \frac{1}{2} \operatorname{tr}(\dot{\Sigma}\Sigma^{-1}\dot{\Sigma}\Sigma^{-1}).$$

By applying Noether's theorem, we find that

$$J_\Sigma = \Sigma^{-1}\dot{\Sigma}$$

is invariant along the geodesics of $\Theta_{\theta^t,2}$, so they are defined by the equation $\dot{\Sigma} = \Sigma J_\Sigma = \Sigma\Sigma_0^{-1}\dot{\Sigma}_0$ (and therefore, any update preserving the invariant J_Σ will satisfy this first-order differential equation and follow the geodesics of $\Theta_{\theta^t,2}$). The xNES update for the covariance matrix is given by $A(t) = A_0 \exp(tG_M/2)$. Therefore, we have $\Sigma(t) = A_0 \exp(tG_M)A_0^T$, $\Sigma^{-1}(t) = (A_0^{-1})^T \exp(-tG_M)A_0^{-1}$, $\dot{\Sigma}(t) = A_0 \exp(tG_M)G_M A_0^T$ and, finally, $\Sigma^{-1}(t)\dot{\Sigma}(t) = (A_0^{-1})^T G_M A_0^T = \Sigma_0^{-1}\dot{\Sigma}_0$. Therefore, xNES preserves J_Σ, and therefore, xNES follows the geodesics of $\Theta_{\theta^t,2}$ (notice that we had already proven this in Proposition 13, since we are looking at the geodesics of $\mathbb{G}_d$ with a fixed mean). □

Although blockwise GIGO is somewhat "less natural" than GIGO, it can be easier to compute for some splittings (as we have just seen), and in the case of the Gaussian distributions, the mean-covariance splitting seems reasonable.

7. Numerical Experiments

We conclude this article with some numerical experiments to compare the behavior of GIGO, xNES and pure rank-μ CMA-ES (we give the pseudocodes for these algorithms in the Appendix). We made two series of tests. The first one is a performance test, using classical benchmark functions and the settings from [19]. The goal of the second series of tests is to illustrate the computations in Section 6.3 by plotting the trajectories (standard deviation *versus* mean) of these three algorithms in dimension one.

The source code is available at [22].

7.1. Benchmarking

For the first series of experiments, presented in Figure 3, we used the following parameters, taken from [19] (we recall that xNES and pure rank-μ CMA-ES are seen as IGO algorithms):

- Varying dimension.
- Sample size: $\lfloor 4 + 3\log(d) \rfloor$.
- Weights: $w_i = \frac{\max(0,\log(\frac{n}{2}+1)-\log(i))}{\sum_{j=1}^{N} \max(0,\log(\frac{n}{2}+1)-\log(j))} - \frac{1}{N}$.
- IGO step size and learning rates: $\delta t = 1, \eta_\mu = 1, \eta_\Sigma = \frac{3}{5}\frac{3+\log(d)}{d\sqrt{d}}$..
- Initial position: $\theta^0 = \mathcal{N}(x_0, I)$, where x_0 is a random point of the circle with center zero, and radius 10.
- Euler method for GIGO: Number of steps: 100. We used the GIGO-A variant of the algorithm. No significant difference was noticed with GIGO-Σ or with the exact GIGO algorithm. The only advantage of having an explicit solution of the geodesic equations is that the update is quicker to compute.
- We chose not to use the exact expression of the geodesics for this benchmarking to show that having to use the Euler method is fine. However, we did run the tests, and the results are basically the same as GIGO-A.

We plot the median number of runs to achieve target fitness (10^{-8}). Each algorithm has been tested in dimension 2, 4, 8, 16, 32 and 64: a missing point means that all runs converged prematurely.

7.1.1. Failed Runs

In Figure 3, a point is plotted even if only one run was successful. Below is the list of the settings for which at least one run converged prematurely.

- Only one run reached the optimum for the cigar-tablet function with CMA-ES in dimension eight.
- Seven runs (out of 24) reached the optimum for the Rosenbrock function with CMA-ES in dimension 16.
- About half of the runs reached the optimum for the sphere function with CMA-ES in dimension four.

Dimension	d	From 2 to 64
Sample size	N	$4 + 3\log(d)$
Weights	$(w_i)_{i\in[1,N]}$	$\frac{\max(0,\log(\frac{n}{2}+1)-\log(i))}{\sum_{j=1}^{N}\max(0,\log(\frac{n}{2}+1)-\log(j))} - \frac{1}{N}$
IGO step size	δt	1
Mean learning rate	η_μ	1
Covariance learning rate	η_Σ	$\frac{3}{5}\frac{3+\log(d)}{d\sqrt{d}}$
Euler step-size (for GIGO only)	h	0.01(100 steps)
GIGO implementation		GIGO-A
Sphere function		$x \mapsto \sum_{i=1}^{d} x_i^2$
Cigar-tablet		$x \mapsto x_1^2 + \sum_{i=2}^{d-1} 10^4 x_i^2 + 10^8 x_d^2$
Rosenbrock		$x \mapsto \sum_{i=1}^{d-1}(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$
x-axis		Dimension
y-axis		Number of function calls to reach fitness 10^{-8}.

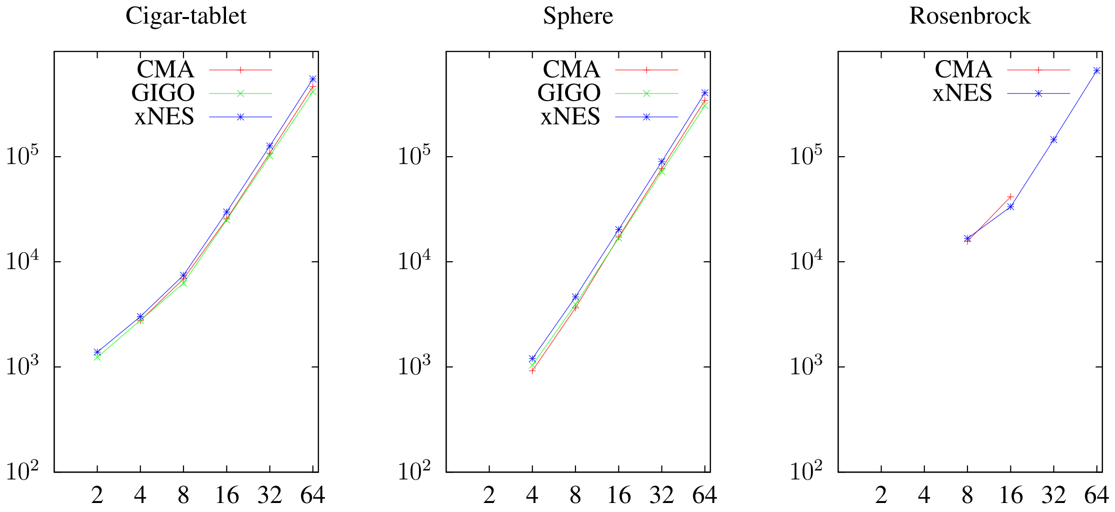

Figure 3. Median number of function calls to reach 10^{-8} fitness on 24 runs for: sphere function, cigar-tablet function and Rosenbrock function. Initial position $\theta^0 = \mathcal{N}(x_0, I)$, with x_0 uniformly distributed on the circle of center zero and radius 10. We recall that the "CMA-ES" algorithm here is using the so-called pure rank-μ CMA-ES update.

For the following settings, all runs converged prematurely.

- GIGO did not find the optimum of the Rosenbrock function in any dimension.
- CMA-ES did not find the optimum of the Rosenbrock function in dimension 2, 4, 32 and 64.
- All of the runs converged prematurely for the cigar-tablet function in dimension two with CMA-ES, for the sphere function in dimension two for all algorithms and for the Rosenbrock function in dimension two and four for all algorithms.

7.1.2. Discussion

As the last item in Section 7.1.1 shows, all of the algorithms converge prematurely in a low dimension, probably because the covariance learning rate has been set too high (or because the sample size is too small). This is different from the results in [19].

This remark aside, as noted in [19], the xNES algorithm shows more robustness than CMA-ES and GIGO: it is the only algorithm able to find the minimum of the Rosenbrock function in high dimensions. However, its convergence is consistently slower.

In terms of performance, when both of them work, pure rank-μ CMA-ES (or equivalently, IGO in the parametrization (μ, Σ)) and GIGO are extremely close (GIGO is usually a bit better). An advantage of GIGO is that it is theoretically defined for any δt, η_Σ, whereas the covariance matrix maintained by CMA-ES (not only pure rank-μ CMA-ES) can stop being positive definite if $\eta_\Sigma \delta t > 1$. However, in that case, the GIGO algorithm is prone to premature convergence (remember Figure 2 and see Proposition 15 below), and in practice, the learning rates are much smaller.

7.2. Plotting Trajectories in $\mathbb{G}_1$

We want the second series of experiments to illustrate the remarks about the trajectories of the algorithms in Section 6.3, so we decided to take a large sample size to limit randomness, and we chose a fixed starting point for the same reason. We use the weights below because of the property of quantile improvement proven in [23]: the $1/4$-quantile will improve at each step. The parameters we used were the following:

- Sample size: $\lambda = 5,000$
- Dimension one only.
- Weights: $w = 4\mathbf{1}_{q \leqslant 1/4}$ ($w_i = 4.\mathbf{1}_{i \leqslant 1{,}250}$)
- IGO step size and learning rates: $\eta_\mu = 1, \eta_\Sigma = \frac{3}{5}\frac{3+\log(d)}{d\sqrt{d}} = 1.8$, varying δt.
- Initial position: $\theta^0 = \mathcal{N}(10, 1)$
- Dots are placed at t = 0, 1, 2 . . . (except for the graph $\delta t = 1.5$, for which there is a dot for each step).

Figures 4–8 show the optimization of $x \mapsto x^2$, and Figures 9–11 show the optimization of $x \mapsto -x$.

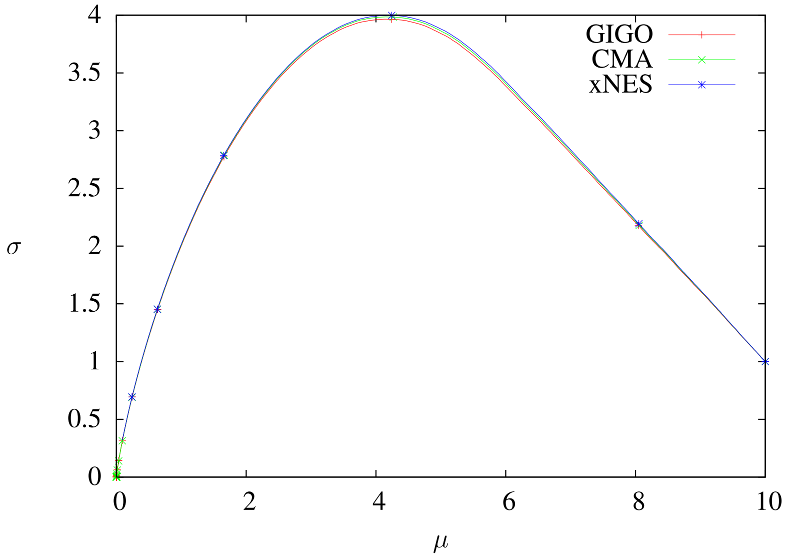

Figure 4. Trajectories of GIGO, CMA and xNES optimizing $x \mapsto x^2$ in dimension one with $\delta t = 0.01$, sample size 5000, weights $w_i = 4.\mathbf{1}_{i \leqslant 1250}$ and learning rates $\eta_\mu = 1$, $\eta_\Sigma = 1.8$. One dot every 100 steps. All algorithms exhibit a similar behavior

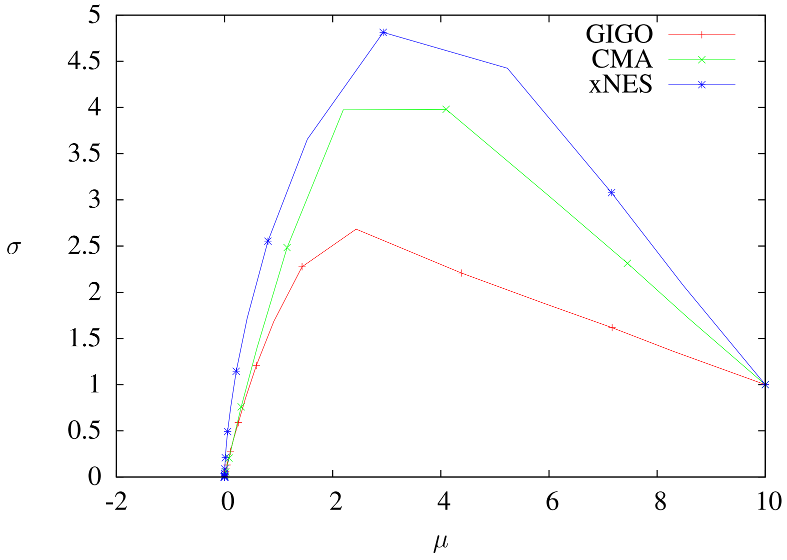

Figure 5. Trajectories of GIGO, CMA and xNES optimizing $x \mapsto x^2$ in dimension one with $\delta t = 0.5$, sample size 5000, weights $w_i = 4.\mathbf{1}_{i \leqslant 1250}$ and learning rates $\eta_\mu = 1$, $\eta_\Sigma = 1.8$. One dot every two steps. Stronger differences. Notice that after one step, the lowest mean is still GIGO (~ 8.5, whereas xNES is around 8.75), but from the second step, GIGO has the highest mean, because of the lower variance.

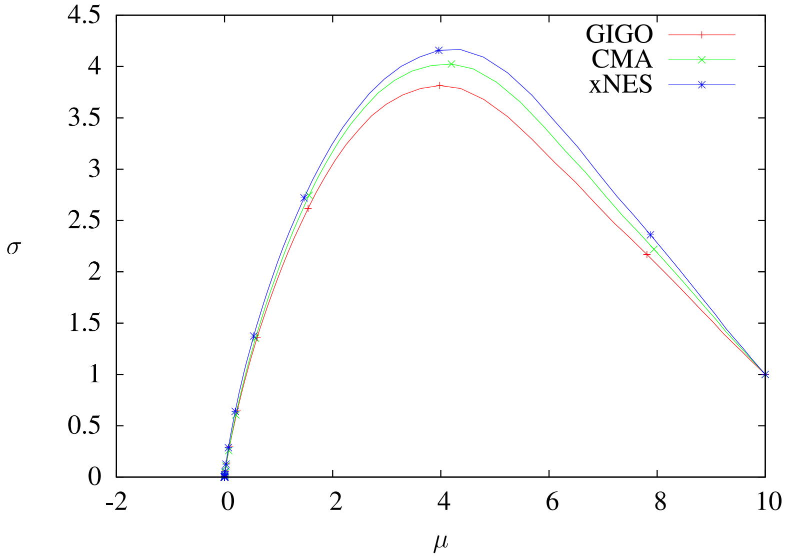

Figure 6. Trajectories of GIGO, CMA and xNES optimizing $x \mapsto x^2$ in dimension one with $\delta t = 0.1$, sample size 5000, weights $w_i = 4.\mathbf{1}_{i \leqslant 1250}$ and learning rates $\eta_\mu = 1$, $\eta_\Sigma = 1.8$. One dot every 10 steps. All algorithms exhibit a similar behavior, and differences start to appear. It cannot be seen on the graph, but the algorithm closest to zero after 400 steps is CMA ($\sim 1.10^{-16}$, followed by xNES ($\sim 6.10^{-16}$) and GIGO ($\sim 2.10^{-15}$).

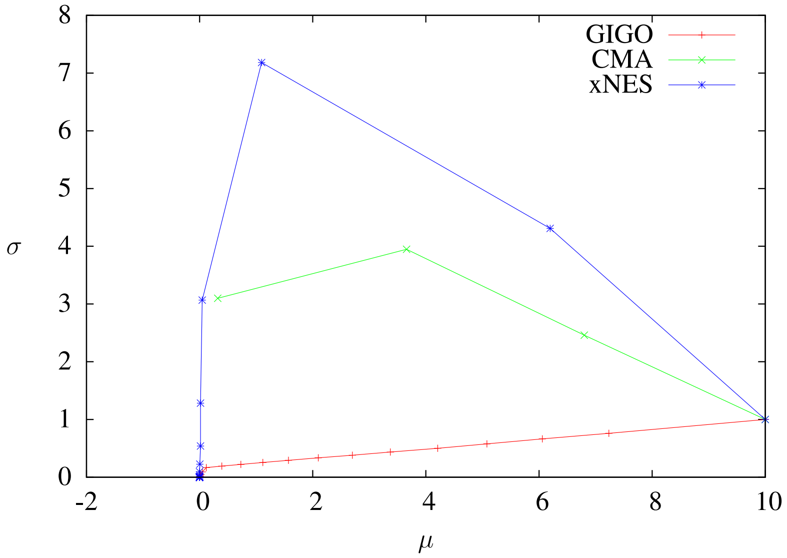

Figure 7. Trajectories of GIGO, CMA and xNES optimizing $x \mapsto x^2$ in dimension one with $\delta t = 1$, sample size 5000, weights $w_i = 4.\mathbf{1}_{i \leqslant 1250}$ and learning rates $\eta_\mu = 1$, $\eta_\Sigma = 1.8$. One dot per step. The CMA-ES algorithm fails here, because at the fourth step, the covariance matrix is not positive definite anymore (it is easy to see that the CMA-ES update is always defined if $\delta t \eta_\Sigma < 1$, but this is not the case here). Furthermore, notice (see also Proposition 15) that at the first step, GIGO decreases the variance, whereas the σ-component of the IGO speed is positive.

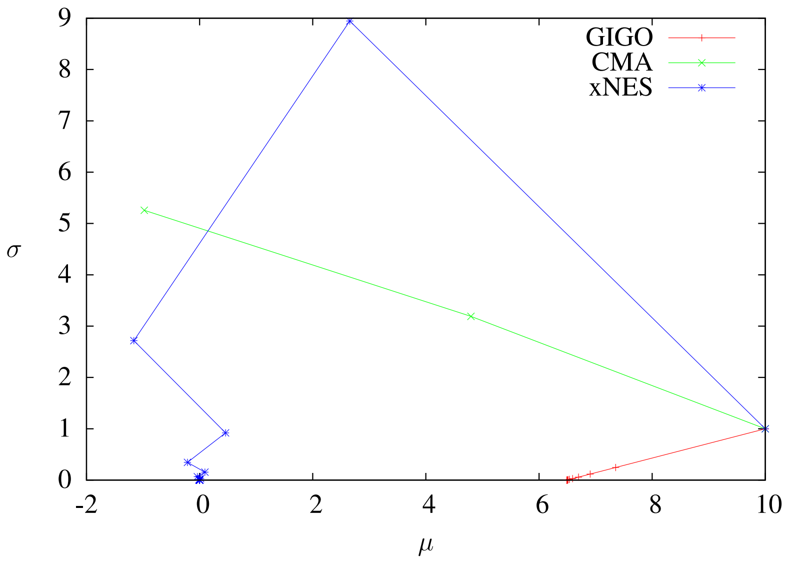

Figure 8. Trajectories of GIGO, CMA and xNES optimizing $x \mapsto x^2$ in dimension one with $\delta t = 1.5$, sample size 5000, weights $w_i = 4.\mathbf{1}_{i \leqslant 1250}$ and learning rates $\eta_\mu = 1$, $\eta_\Sigma = 1.8$. One dot per step. Same as $\delta t = 1$ for CMA. GIGO converges prematurely.

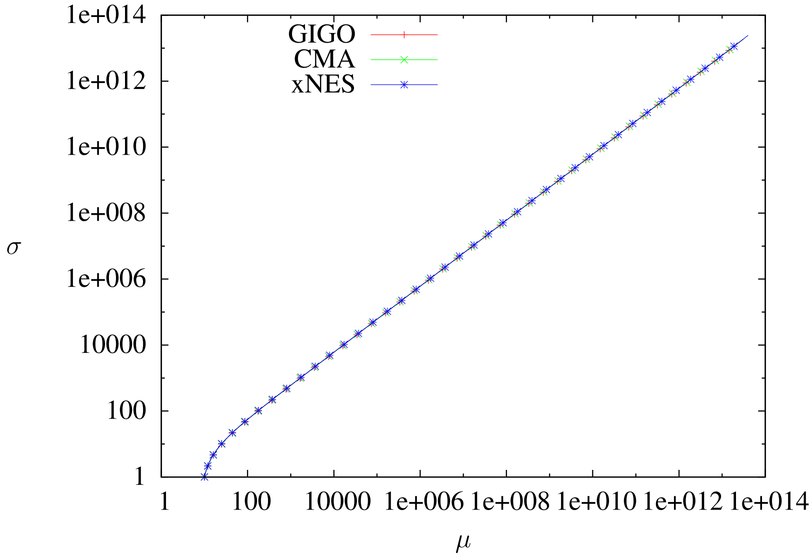

Figure 9. Trajectories of GIGO, CMA and xNES optimizing $x \mapsto -x$ in dimension one with $\delta t = 0.01$, sample size 5000, weights $w_i = 4.\mathbf{1}_{i \leqslant 1250}$ and learning rates $\eta_\mu = 1$, $\eta_\Sigma = 1.8$. One dot every 100 steps. Almost the same for all algorithms.

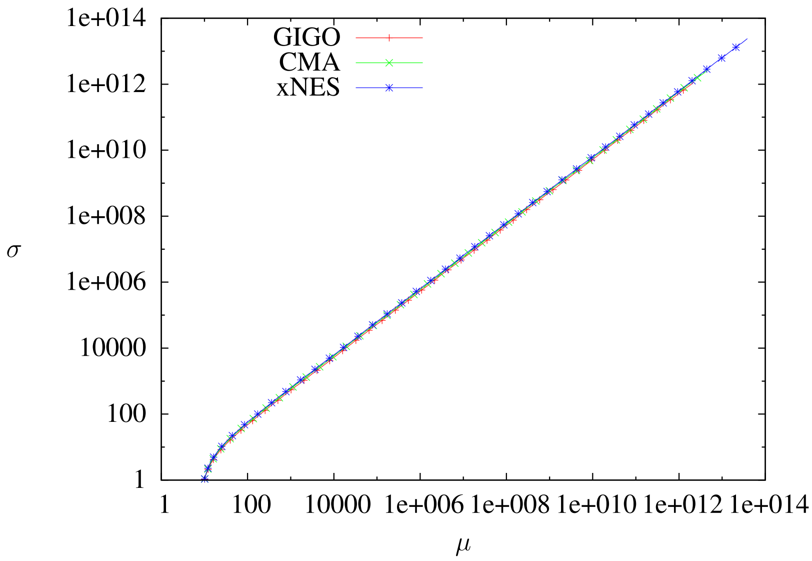

Figure 10. Trajectories of GIGO, CMA and xNES optimizing $x \mapsto -x$ in dimension one with $\delta t = 0.1$, sample size 5000, weights $w_i = 4.\mathbf{1}_{i \leqslant 1250}$ and learning rates $\eta_\mu = 1$, $\eta_\Sigma = 1.8$. One dot every 10 steps. It is not obvious on the graph, but xNES is faster than CMA, which is faster than GIGO.

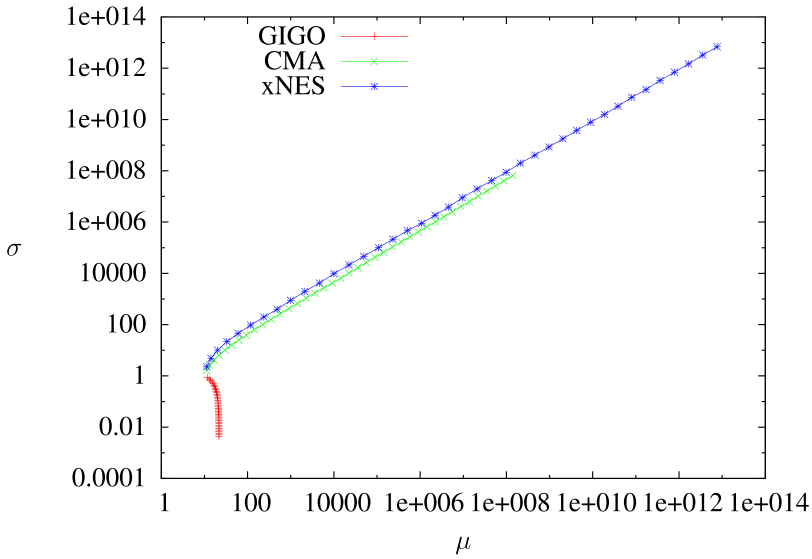

Figure 11. Trajectories of GIGO, CMA and xNES optimizing $x \mapsto -x$ in dimension one with $\delta t = 1$, sample size $5,000$, weights $w_i = 4.\mathbf{1}_{i \leqslant 1,250}$ and learning rates $\eta_\mu = 1$, $\eta_\Sigma = 1.8$. One dot per step. GIGO converges, for the reasons discussed earlier.

Figures 7, 8 and 11 show that when $\delta t \geqslant 1$, GIGO reduces the covariance, even at the first step. More generally, when using the GIGO algorithm in $\tilde{\mathbb{G}}_d$ for the optimization of a linear function, there exists a critical step size δt_{cr} (depending on the learning rates η_μ, η_σ and on the weights w_i), above which, GIGO will converge, and we can compute its value when the weights are of the form $\mathbf{1}_{q \leqslant q_0}$ (for $q_0 \geqslant 0.5$, the discussion is not relevant, because in that case, even the IGO flow converges prematurely. Compare with the critical δt of the smoothed cross entropy method and IGO-ML in [1]).

Proposition 15. *Let $d \in \mathbb{N}$, k, η_μ, $\eta_\sigma \in \mathbb{R}_+^*$; let $w = k.\mathbf{1}_{q \leqslant q_0}$; and let*

$$\begin{array}{cccc} g & : & \mathbb{R}^d & \to & \mathbb{R} \\ & & x & \mapsto & -x_1 \end{array}.$$

Let μ_n be the first coordinate of the mean, and let σ_n^2 be the variance (at step n) maintained by the (η_μ, η_σ)-twisted geodesic IGO algorithm in $\tilde{\mathbb{G}}_d$ associated with selection scheme w, sample size ∞ and step size δt, when optimizing g ("sample size ∞" meaning the limit of the update when the sample size tends to infinity, which is deterministic [1]).

There exists δt_{cr}, such that:

- *if $\delta t > \delta t_{cr}$, (σ_n) converges to zero with exponential speed and (μ_n) converges.*
- *if $\delta t = \delta t_{cr}$, (σ_n) remains constant and (μ_n) tends to ∞ with linear speed.*
- *if $0 < \delta t < \delta t_{cr}$, both (σ_n) and μ_n tend to ∞ with exponential speed.*

The proof and the expression of δt_{cr} can be found in the Appendix.

In the case corresponding to $k = 4$, $n = 1$, $q_0 = 1/4$, $\eta_\mu = 1$, $\eta_\sigma = 1.8$, we find:

$$\delta t_{\text{cr}} \approx 0.84. \tag{64}$$

8. Conclusions

We introduced the geodesic IGO algorithm, and we showed that in the case of Gaussian distributions, Noether's theorem directly gives a first order equation satisfied by the geodesics. In terms of performance, the GIGO algorithm is similar to pure rank-μ CMA-ES, which is rather encouraging: it would be interesting to test GIGO on real problems. Moreover, GIGO is a reasonable and totally parametrization-invariant algorithm (provided we can compute the solution of the equations of the geodesics), and as such, it should be studied for other families of probability distributions, like Bernoulli distributions (although in that case, the Riemannian exponential is not defined if the step size is too large, because the length of the geodesics is finite). Noether's theorem could be a crucial tool for this.

We also showed that xNES and GIGO are not the same algorithm, and we defined blockwise GIGO, a simple extension of the GIGO algorithm, showing that xNES has a special status, as it admits a definition that is "almost" parametrization-invariant.

Acknowledgments

I would like to thank Yann Ollivier for his numerous remarks about this article and Frédéric Barbaresco for finding [5].

Appendix A

Proof of Proposition 15. Let us first consider the case $k = 1$.

When optimizing a linear function, the non-twisted IGO flow in $\tilde{\mathbb{G}}_d$ with the selection function $w : q \mapsto \mathbf{1}_{q \leqslant q_0}$ is known [1], and in particular, we have:

$$\mu_t = \mu_0 + \frac{\beta(q_0)}{\alpha(q_0)}\sigma_t, \tag{65}$$

$$\sigma_t = \sigma_0 \exp(\alpha(q_0)t), \tag{66}$$

where, if we denote by $\mathcal{N}$ a random vector following a standard normal distribution and $\mathcal{F}$ the cumulative distribution of a standard normal distribution,

$$\alpha(q_0, d) = \frac{1}{2d}\left(\int_0^{q_0} \mathcal{F}^{-1}(u)^2 du - q_0\right), \tag{67}$$

and:

$$\beta(q_0) = \mathbb{E}(\mathcal{N}\mathbf{1}_{\mathcal{N}\leqslant \mathcal{F}^{-1}(q_0)}). \tag{68}$$

In particular, $\alpha := \alpha(\frac{1}{4}, 1) \approx 0.107$ and $\beta := \beta(\frac{1}{4}) \approx -0.319$.

With a minor modification of the proof in [1], we find that the (η_μ, η_σ)-twisted IGO flow is given by:

$$\mu_t = \mu_0 + \frac{\beta(q_0)}{\alpha(q_0)}\sigma_0 \exp(\eta_\mu \alpha(q_0)t), \tag{69}$$

$$\sigma_t = \sigma_0 \exp(\eta_\sigma \alpha(q_0)t), \tag{70}$$

Notice that Equation (69) shows that the assertions about the convergence of (σ_n) immediately imply the assertions about the convergence of (μ_n).

Let us now consider a step of the GIGO algorithm: The twisted IGO speed is $Y = (\eta_\mu\beta\sigma_0, \eta_\sigma\alpha\sigma_0)$, with $\alpha\sigma_0 > 0$ (*i.e.*, the variance should be increased: this is where we need $q_0 < 0.5$).

Proposition 17 shows that the covariance at the end of the step is (using the same notation):

$$\sigma(\delta t) = \sigma(0)\mathrm{Im}(\frac{die^{v\delta t} - c}{cie^{v\delta t} + d}) = \sigma(0)\frac{e^{v\delta t}(d^2 + c^2)}{c^2e^{2v\delta t} + d^2} =: \sigma(0)f(\delta t), \tag{71}$$

and it is easy to see that f only depends on δt (and on q_0). In other words, $f(\delta t)$ will be the same at each step of the algorithm. The existence of δt_{cr} easily follows (furthermore, recall Figure 1 in Section 4.1), and δt_{cr} is the positive solution of $f(x) = 1$.

After a quick computation, we find:

$$\exp(v\delta t_{\mathrm{cr}}) = \frac{\sqrt{1+u^2}+1}{\sqrt{1+u^2}-1}. \tag{72}$$

where:

$$u := \sqrt{\frac{\eta_\mu}{2n\eta_\sigma}}\frac{\beta}{\alpha}, \tag{73}$$

and:

$$v := \sqrt{\eta_\sigma^2\alpha^2 + \frac{\eta_\mu\eta_\sigma}{2n}\beta^2}. \tag{74}$$

Finally, for $w = k.\mathbf{1}_{q\leqslant q_0}$, Proposition 9 shows that:

$$\delta t_{\mathrm{cr}} = \frac{1}{k}\frac{1}{v}\ln\left(\frac{\sqrt{1+u^2}+1}{\sqrt{1+u^2}-1}\right). \tag{75}$$

□

A1. Generalization of the Twisted Fisher Metric

The following definition is a more general way to introduce the twisted Fisher metric.

Definition 17. *Let (Θ, g) be a Riemannian manifold, $(\Theta_1, g|_{\Theta_1}), ..., (\Theta_n, g|_{\Theta_n})$, a splitting (as defined in Section 6.4) of Θ compatible with the metric g.*

We call $(\eta_1, ..., \eta_n)$-twisted metric on (Θ, g) for the splitting $\Theta_1, ..., \Theta_n$ the metric g' on Θ defined by $g'|_{\Theta_i} = \frac{1}{\eta_i}g|_{\Theta_i}$ for $1 \leqslant i \leqslant n$, and $\Theta_i \perp \Theta_j$ for $i \neq j$.

Proposition 16. *The (η_μ, η_Σ)-twisted metric on $\mathbb{G}_d$ with the Fisher metric for the splitting $\mathcal{N}(\mu, \Sigma) \mapsto (\mu, \Sigma)$ coincides with the (η_μ, η_Σ)-twisted Fisher metric from Definition 11.*

Proof. It is easy to see that the (η_μ, η_Σ)-twisted Fisher metric satisfies the condition in Definition 17. □

A2. Twisted Geodesics

The following theorem can be used to compute the twisted geodesics from the non twisted geodesics. It is a simple calculation.

Theorem 7. *Let $\eta_\mu, \eta_\Sigma \in \mathbb{R}$, $\mu_0 \in \mathbb{R}^d$, $A_0 \in GL_d(\mathbb{R})$, and $(\dot{\mu}_0, \dot{\Sigma}_0) \in T_{\mathcal{N}(\mu_0, A_0A_0^T)}\mathbb{G}_d$. Let*

$$\begin{array}{cccc} h & : & \mathbb{G}_d & \to & \mathbb{G}_d \\ & & \mathcal{N}(\mu, \Sigma) & \mapsto & \mathcal{N}(\sqrt{\frac{\eta_\mu}{\eta_\Sigma}}\mu, \Sigma) \end{array}. \tag{76}$$

We denote by ϕ (resp. ψ) the Riemannian exponential of $\mathbb{G}_d$ (resp. $\mathbb{G}_d$ with the (η_μ, η_Σ)-twisted Fisher metric) at $\mathcal{N}(\sqrt{\frac{\eta_\mu}{\eta_\Sigma}}\mu_0, A_0A_0^T)$ (resp. $\mathcal{N}(\mu_0, A_0A_0^T)$). We have:

$$\psi(\dot{\mu}_0, \dot{\Sigma}_0) = h \circ \phi(\sqrt{\frac{\eta_\Sigma}{\eta_\mu}}\dot{\mu}_0, \dot{\Sigma}_0) \tag{77}$$

Proof. Let us denote by: $\begin{pmatrix} I_\mu & 0 \\ 0 & I_\Sigma \end{pmatrix}$ the Fisher metric in the parametrization μ, Σ, and consider the following parametrization of $\mathbb{G}_d$: $(\tilde{\mu}, \Sigma) \mapsto \mathcal{N}(\frac{\sqrt{\eta_\Sigma}}{\sqrt{\eta_\mu}}\tilde{\mu}, \Sigma)$.

The Riemannian exponential at $\mathcal{N}(\mu_0, A_0A_0^T)$ in this parametrization is:

$$h \circ \phi \circ (\mathrm{d}h(\mu_0, A_0A_0^T))^{-1} \tag{78}$$

However, in this parametrization, the Fisher metric reads:

$$\begin{pmatrix} \frac{\eta_\Sigma}{\eta_\mu} I_\mu & 0 \\ 0 & I_\Sigma \end{pmatrix}, \tag{79}$$

which is proportional to the (η_μ, η_Σ)-twisted Fisher metric up to a factor $\frac{1}{\eta_\Sigma}$. Consequently, the Christoffel symbols are the same as the Christoffel symbols of the (η_μ, η_Σ)-twisted Fisher metric, and so are the geodesics. Therefore, we have:

$$\psi = h \circ \phi \circ (\mathrm{d}h(\mu_0, A_0 A_0^T))^{-1}, \tag{80}$$

which is what we wanted. □

For the remainder of this section, we fix η_μ and η_Σ; $\mathbb{G}_d$ is endowed with the (η_μ, η_Σ)-twisted Fisher metric, and $\tilde{\mathbb{G}}_d$ is endowed with the induced metric. The proofs of the propositions below are a simple rewriting of their non-twisted counterparts that can be found in Sections 4 and 5.1 and can be seen as corollaries of Theorem 7.

Theorem 8. *If* $\gamma\colon t \mapsto \mathcal{N}(\mu(t), \sigma(t)^2 I)$ *is a twisted geodesic of* $\tilde{\mathbb{G}}_d$*, then there exists* $a, b, c, d \in \mathbb{R}$*, such that* $ad - bc = 1$*, and* $v > 0$*, such that*
$\mu(t) = \mu(0) + \sqrt{\frac{2d\eta_\mu}{\eta_\sigma}} \frac{\dot{\mu}_0}{\|\dot{\mu}_0\|} \tilde{r}(t)$, $\sigma(t) = \mathrm{Im}(\gamma_{\mathbb{C}}(t))$*, with* $\tilde{r}(t) = \mathrm{Re}(\gamma_{\mathbb{C}}(t))$ *and:*

$$\gamma_{\mathbb{C}}(t) := \frac{aie^{vt} + b}{cie^{vt} + d}. \tag{81}$$

Proposition 17. *Let* $n \in \mathbb{N}$*,* $v_\mu \in \mathbb{R}^n$*,* v_σ*,* η_μ*,* η_σ*,* $\sigma_0 \in \mathbb{R}$*, with* $\sigma_0 > 0$*.*
Let $v_r := \|v_\mu\|$*,* $\lambda = \sqrt{\frac{2n\eta_\mu}{\eta_\sigma}}$ $v := \sqrt{\frac{\frac{1}{\lambda^2} v_r^2 + v_\sigma^2}{\sigma_0^2}}$*,* $M_0 := \frac{1}{\lambda} \frac{v_r}{v\sigma_0^2}$ *and* $S_0 := \frac{v_\sigma}{v\sigma_0^2}$*.*
Let $c := \left(\frac{\sqrt{M_0^2 + S_0^2} - S_0}{2} \right)^{\frac{1}{2}}$ *and* $d := \left(\frac{\sqrt{M_0^2 + S_0^2} + S_0}{2} \right)^{\frac{1}{2}}$*.*
Let $\gamma_{\mathbb{C}}(t) := \sigma_0 \frac{die^{vt} - c}{cie^{vt} + d}$*.*
Then:

$$\gamma : t \mapsto \mathcal{N}\left(\mu_0 + \lambda \frac{v_\mu}{\|v_\mu\|} \mathrm{Re}(\gamma_{\mathbb{C}}(t)), \mathrm{Im}(\gamma_{\mathbb{C}}(t)) \right) \tag{82}$$

is the twisted geodesic of $\tilde{\mathbb{G}}_n$ *satisfying* $\gamma(0) = (\mu_0, \sigma_0)$ *and* $\dot{\gamma}(0) = (v_\mu, v_\sigma)$*. The regular geodesics of* $\tilde{\mathbb{G}}_n$ *are obtained with* $\eta_\mu = \eta_\sigma = 1$*.*

Theorem 9. *Let* $\gamma : t \mapsto \mathcal{N}(\mu_t, \Sigma_t)$ *be a twisted geodesic of* $\mathbb{G}_d$*. Then, the following quantities are invariant:*

$$J_\mu = \frac{1}{\eta_\mu} \Sigma_t^{-1} \dot{\mu}_t, \tag{83}$$

$$J_\Sigma = \Sigma_t^{-1} \left(\frac{1}{\eta_\mu} \dot{\mu}_t \mu_t^T + \frac{1}{\eta_\Sigma} \dot{\Sigma}_t\right). \tag{84}$$

Theorem 10. *If* $\mu : t \mapsto \mu_t$ *and* $\Sigma : t \mapsto \Sigma_t$ *satisfy the equations:*

$$\dot{\mu}_t = \eta_\mu \Sigma_t J_\mu \tag{85}$$

$$\dot{\Sigma}_t = \eta_\Sigma \Sigma_t (J_\Sigma - J_\mu \mu_t^T) = \eta_\Sigma \Sigma_t J_\Sigma - \frac{\eta_\Sigma}{\eta_\mu} \dot{\mu}_t \mu_t^T, \tag{86}$$

where:

$$J_\mu = \frac{1}{\eta_\mu} \Sigma_0^{-1} \dot{\mu}_0,$$

and:

$$J_\Sigma = \Sigma_0^{-1} \left(\frac{1}{\eta_\mu} \dot{\mu}_0 \mu_0^T + \frac{1}{\eta_\Sigma} \dot{\Sigma}_0 \right).$$

then $t \mapsto \mathcal{N}(\mu_t, \Sigma_t)$ is a twisted geodesic of $\mathbb{G}_d$.

Theorem 11. *If $\mu : t \mapsto \mu_t$ and $A : t \mapsto A_t$ satisfy the equations:*

$$\dot{\mu} = \eta_\mu A_t A_t^T J_\mu, \tag{87}$$

$$\dot{A}_t = \frac{\eta_\Sigma}{2} (J_\Sigma - J_\mu \mu_t^T)^T A_t, \tag{88}$$

where:

$$J_\mu = \frac{1}{\eta_\mu} (A_0^{-1})^T A_0^{-1} \dot{\mu}_0$$

and:

$$J_\Sigma = (A_0^{-1})^T A_0^{-1} \left(\frac{1}{\eta_\mu} \dot{\mu}_0 \mu_0^T + \frac{1}{\eta_\Sigma} \dot{A}_0 A_0^T + \frac{1}{\eta_\Sigma} A_0 \dot{A}_0^T \right),$$

then $t \mapsto \mathcal{N}(\mu_t, A_t A_t^T)$ is a twisted geodesic of $\mathbb{G}_d$.

A3. Pseudocodes

A3.1. For All Algorithms

All studied algorithms have a common part, given here:

Variables: μ, Σ (or A such that $\Sigma = AA^T$).

List of parameters: $f \colon \mathbb{R}^d \to \mathbb{R}$, step size δt, learning rates η_μ, η_Σ, sample size λ, weights $(w_i)_{i \in [1,\lambda]}$, N number of steps for the Euler method, r Euler step size reduction factor (for GIGO-Σ only).

Algorithm 1 For all algorithms.

```
$\mu \leftarrow \mu_0$
**if** The algorithm updates $\Sigma$ directly **then**
    $\Sigma \leftarrow \Sigma_0$
    Find some $A$, such that $\Sigma = AA^T$
**else** {The algorithm updates a square root $A$ of $\Sigma$}
    $A \leftarrow A_0$
    $\Sigma = AA^T$
**end if**
**while** NOT (Termination criterion) **do**
    **for** $i = 1$ to $\lambda$ **do**
        $z_i \sim \mathcal{N}(0, I)$
        $x_i = Az_i + \mu$
    **end for**
    Compute the IGO initial speed, and update the mean and the covariance (the updates are Algorithms 2 to 6).
**end while**
```

Notice that we always need a square root A of Σ to sample the x_i, but the decomposition $\Sigma = AA^T$ is not unique. Two different decompositions will give two algorithms, such that one is a modification of the other as a stochastic process: same law (the x_i are abstractly sampled from $\mathcal{N}(\mu, \Sigma)$), but different trajectories (for given z_i, different choices for the square root will give different x_i). For GIGO-Σ, since we have to invert the covariance matrix, we used the Cholesky decomposition (A lower triangular. The the other implementation directly maintains a square root of Σ). Usually, in CMA-ES, the square root of Σ ($\Sigma = AA^T$, A symmetric) is used.

A3.2. Updates

When describing the different updates, μ, Σ, A, the x_i and the z_i are those defined in Algorithm 1.

For Algorithm 2 (GIGO-Σ), when the covariance matrix after one step is not positive-definite, we compute the update again, with a step size divided by r for the Euler method (we have no reason to recommend any particular value of r, the only constraint is $r > 1$).

Algorithm 2 GIGO Update, one step, updating the covariance matrix.

```
1. Compute the IGO speed:
$v_\mu = A \sum_{i=1}^{\lambda} w_i z_i,$
$v_\Sigma = A \sum_{i=1}^{\lambda} w_i \left( z_i z_i^T - I \right) A^T.$
2. Compute the Noether invariants:
$J_\mu \leftarrow \Sigma^{-1} v_\mu,$
$J_\Sigma \leftarrow \Sigma^{-1}(v_\mu{}^t\mu + v_\Sigma).$
3. Solve numerically the equations of the geodesics:
Unhappy $\leftarrow$ true
$\mu_0 \leftarrow \mu$
$\Sigma_0 \leftarrow \Sigma$
$k = 0$
while Unhappy do
    $\mu \leftarrow \mu_0$
    $\Sigma \leftarrow \Sigma_0$
    $h \leftarrow \delta t/(Nr^k)$
    for $i = 1$ to $Nr^k$ do
        $\mu \leftarrow \mu + h\eta_\mu \Sigma J_\mu$
        $\Sigma \leftarrow \Sigma + h\eta_\Sigma \Sigma(J_\Sigma - J_\mu \mu^T)$
    end for
    if $\Sigma$ positive-definite then
        Unhappy $\leftarrow$ false
    end if
    $k \leftarrow k + 1$
end while
return $\mu, \Sigma$
```

Algorithm 3 GIGO Update, one step, updating a square root of the covariance matrix.

```
1. Compute the IGO speed:
$v_\mu = A \sum_{i=1}^{\lambda} w_i z_i,$
$v_\Sigma = A \sum_{i=1}^{\lambda} w_i \left( z_i z_i^T - I \right) A^T.$
2. Compute the Noether invariants:
$J_\mu \leftarrow \Sigma^{-1} v_\mu,$
$J_\Sigma \leftarrow \Sigma^{-1}(v_\mu{}^t\mu + v_\Sigma).$
3. Solve numerically the equations of the geodesics:
$h \leftarrow \delta t / N$
for $i = 1$ to $N$ do
    $\mu \leftarrow \mu + h\eta_\mu A A^T J_\mu$
    $A \leftarrow A + \frac{h}{2}\eta_\Sigma (J_\Sigma - J_\mu \mu^T)^T A$
end for
return $\mu$, $A$
```

Algorithm 4 Exact GIGO, one step. Not exactly our implementation; see the discussion after Corollary 1.

```
1. Compute the IGO speed:
$v_\mu = A \sum_{i=1}^{\lambda} w_i z_i,$
$v_\Sigma = A \sum_{i=1}^{\lambda} w_i \left( z_i z_i^T - I \right) A^T.$
2. Learning rates
$\lambda \leftarrow \sqrt{\frac{\eta_\Sigma}{\eta_\mu}}$
$\mu \leftarrow \lambda \mu$
$v_\mu \leftarrow \eta_\mu \lambda v_\mu$
$v_\Sigma \leftarrow \eta_\Sigma v_\Sigma$
3. Intermediate computations.
$G^2 \leftarrow A^{-1} \left( v_\Sigma (A^{-1})^T A^{-1} v_\Sigma + 2 v_\mu v_\mu^T \right) (A^{-1})^T$
$C_1 \leftarrow \mathrm{ch}(\frac{G}{2})$
$C_2 \leftarrow \mathrm{sh}(\frac{G}{2}) G^{-1}$
$R \leftarrow \left( (C_1 - A^{-1} v_\Sigma (A^{-1})^T C_2)^{-1} \right)^T$
4. Actual update
$\mu \leftarrow \mu + 2ARC_2A^{-1}v_\mu$
$A \leftarrow AR$
5. Return to the "real" $\mu$
$\mu \leftarrow \frac{\mu}{\lambda}$
return $\mu$, $A$
```

Algorithm 5 xNES update, one step.

```
1. Compute $G_\mu$ and $G_M$ (equivalent to the computation of the IGO speed):
$G_\mu = \sum_{i=1}^{\lambda} w_i z_i$
$G_M = \sum_{i=1}^{\lambda} w_i \left( z_i z_i^T - I \right)$
2. Actual update:
$\mu \leftarrow \mu + \eta_\mu A G_\mu$
$A \leftarrow A + A \exp(\eta_\Sigma G_M / 2)$

return $\mu$, $A$
```

Algorithm 6 pure rank-μ CMA-ES update, one step

```
1. Computation of the IGO speed:
$v_\mu = \sum_{i=1}^{\lambda} w_i (x_i - \mu)$
$v_\Sigma = \sum_{i=1}^{\lambda} w_i \left( (x_i - \mu)(x_i - \mu)^T - \Sigma \right)$
2. Actual update:
$\mu \leftarrow \mu + \eta_\mu v_\mu$
$\Sigma \leftarrow \Sigma + \eta_\Sigma v_\Sigma$
return $\mu$, $\Sigma$
```

Algorithm 7 GIGO in $\tilde{\mathbb{G}}_d$, one step.

```
1. Compute the IGO speed:
$Y_\mu = \sum_{i=1}^{\lambda} w_i (x_i - \mu) \ ; \ Y_\sigma = \sum_{i=1}^{\lambda} w_i \left( \frac{(x_i - \mu)^T (x_i - \mu)}{2d\sigma} - \frac{\sigma}{2} \right)$
2. Better parametrization:
$\lambda := \sqrt{\frac{2d\eta_\mu}{\eta_\sigma}}$
$v_r := \frac{\eta_\mu}{\lambda} \|Y_\mu\| \ ; \ v_\sigma := \eta_\sigma Y_\sigma$
3. Find $a, b, c, d, v$ corresponding to $\mu, \sigma, \dot{\mu}, \dot{\sigma}$:
$v = \sqrt{\frac{v_r^2 + v_\sigma^2}{\sigma^2}}$
$S_0 := \frac{v_\sigma}{v\sigma^2} \ ; \ M_0 := \frac{v_r}{v\sigma^2}$
$C := \frac{\sqrt{S_0^2 + M_0^2} - S_0}{2} \ ; \ D := \frac{\sqrt{S_0^2 + M_0^2} + S_0}{2}$
$c := \sqrt{C} \ ; \ d := \sqrt{D}$
4. Actual Update:
$z := \sigma \frac{die^{v\delta t} - c}{cie^{v\delta t} + d}$
$\mu := \mu + \lambda \mathrm{Re}(z) \frac{Y_\mu}{\|Y_\mu\|} \ ; \ \sigma := \mathrm{Im}(z)$
return $\mu, \sigma$
```

Conflicts of Interest

The author declares no conflict of interest.

References

1. Ollivier, Y.; Arnold, L.; Auger, A.; Hansen, N. Information-geometric optimization algorithms: A unifying picture via invariance principles. **2011**, arXiv:1106.3708.
2. Amari, S.-I.; Nagaoka, H. *Methods of Information Geometry (Translations of Mathematical Monographs)*; American Mathematical Society: Providence, RI, USA, 2007.
3. Malagò, L.; Pistone, G. Combinatorial optimization with information geometry: The Newton method. *Entropy* **2014**, *16*, 4260–4289.
4. Eriksen, P. *Geodesics Connected with the Fisher Metric on the Multivariate Normal Manifold*; Technical Report 86-13; Institute of Electronic Systems, Aalborg University: Aalborg, Denmark, 1986.
5. Calvo, M.; Oller, J.M. An Explicit Solution of Information Geodesic Equations for the Multivariate Normal Model. *Stat. Decis.* **1991**, *9*, 119–138.
6. Imai, T.; Takaesu, A.; Wakayama, M. Remarks on geodesics for multivariate normal models. *J. Math-for-Industry* **2011**, *3*, 125–130.
7. Skovgaard, L.T. A Riemannian geometry of the multivariate normal model. *Scand. J. Stat.* **1981**, *11*, 211–223.
8. Porat, B.; Friedlander, B. Computation of the Exact Information Matrix of Gaussian Time Series with Stationary Random Components. *IEEE Trans. Acoust. Speech Signal Process.* **1986**, *34*, 118–130.
9. Baluja, S.; Caruana, R. *Removing the Genetics from the Standard Genetic Algorithm*; Technical Report CMU-CS-95-141; Morgan Kaufmann Publishers: Burlington, MA, USA, 1995, pp. 38–46.
10. Malagò, L.; Matteucci, M.; Pistone, G. Towards the geometry of estimation of distribution algorithms based on the exponential family. In Proceedings of the 11th Workshop Proceedings on Foundations of Genetic Algorithms, Schwarzenberg, Austria, 5–9 January 2011; pp. 230–242.
11. Kern, S.; Müller, S.D.; Hansen, N.; Büche, D.; Ocenasek, J.; Koumoutsakos, P. Learning probability distributions in continuous evolutionary algorithms—A comparative review. *Nat. Comput.* **2003**, *3*, 77–112.
12. Wierstra, D.; Schaul, T.; Glasmachers, T.; Sun, Y.; Peters, J.; Schmidhuber, J. Natural evolution strategies. *J. Mach. Learn. Res.* **2014**, *15*, 949–980.
13. Huang, W. Optimization Algorithms on Riemannian Manifolds with Applications. Ph.D. Thesis, Florida State University, Tallahassee, FL, USA, 2013.
14. Absil, P.A.; Mahony, R.; Sepulchre, R. *Optimization Algorithms on Matrix Manifolds*; Princeton University Press: Princeton, NJ, USA, 2008.
15. Arnold, V.; Vogtmann, K.; Weinstein, A. *Mathematical Methods of Classical Mechanics (Graduate Texts in Mathematics)*; Springer: New York, NY, USA, 1989.
16. Bourguignon, J. *Calcul variationnel*; Ecole Polytechnique: Palaiseau, France, 2007. (in French)

17. Jost, J.; Li-Jost, X. *Calculus of Variations (Cambridge Studies in Advanced Mathematics)*; Cambridge University Press: Cambridge, UK, 1998.
18. Gallot, S.; Hulin, D.; LaFontaine, J. *Riemannian Geometry (Universitext)*, 3rd ed.; Springer: Berlin/Heidelberg, Germany, 2004.
19. Glasmachers, T.; Schaul, T.; Yi, S.; Wierstra, D.; Schmidhuber, J. Exponential natural evolution strategies. In Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, Portland, OR, USA, 7–11 July 2010.
20. Akimoto, Y.; Nagata, Y.; Ono, I.; Kobayashi, S. Bidirectional relation between CMA evolution strategies and natural evolution strategies. In *Parallel Problem Solving from Nature, PPSN XI*; Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G., Eds.; Springer: New York, NY, USA, 2010.
21. Hansen, N. The CMA evolution strategy: A tutorial. Available online: https://www.lri.fr/∼hansen/cmatutorial.pdf (accessed on 1 January 2015).
22. Bensadon, J. Source Code. Available online: https://www.lri.fr/~bensadon/ (accessed on 13 January 2015).
23. Akimoto, Y.; Ollivier, Y. Objective improvement in information-geometric optimization. In Proceedings of the twelfth workshop on Foundations of genetic algorithms XII, Adelaide, Australia, 16–20 January 2013.