# Variable Metric Random Pursuit\*

Sebastian U. Stich<sup> $\dagger$ </sup> C. L. Müller<sup>§</sup>

Bernd Gärtner<sup>‡</sup>

September 30, 2014

#### Abstract

We consider unconstrained randomized optimization of smooth convex objective functions in the gradient-free setting. We analyze Random Pursuit (RP) algorithms with fixed (F-RP) and variable metric (V-RP). The algorithms only use zeroth-order information about the objective function and compute an approximate solution by repeated optimization over randomly chosen one-dimensional subspaces. The distribution of search directions is dictated by the chosen metric. Variable Metric RP uses novel variants of a randomized zeroth-order Hessian approximation scheme recently introduced by Leventhal and Lewis (D. Leventhal and A. S. Lewis., Optimization 60(3), 329-245, 2011). We here present (i) a refined analysis of the expected single step progress of RP algorithms and their global convergence on (strictly) convex functions and (ii) novel convergence bounds for V-RP on strongly convex functions. We also quantify how well the employed metric needs to match the local geometry of the function in order for the RP algorithms to converge with the best possible rate. Our theoretical results are accompanied by numerical experiments, comparing V-RP with the derivative-free schemes CMA-ES, Implicit Filtering, Nelder-Mead, NEWUOA, Pattern-Search and Nesterov's gradient-free algorithms.

# 1 Introduction

Since its inception by Davidon in the late 1950's [6] variable metric methods have become a cornerstone in first-order (non-)convex continuous optimization. Among the many instances of variable metric schemes Quasi-Newton methods such as the BFGS scheme [5, 7, 8, 29] are ubiquitous in all areas of science and engineering. In zeroth-order (or gradient-free) optimization, the idea of using a variable metric guiding the search for local or global optima has surprisingly been used to a far less extent. Although "directional adaptation" has been conjectured to be useful for randomized gradient-free schemes in the late 1960's [28] the literature on this topic is scarce and scattered across different communities ranging from electrical engineering, optimal control, bio-inspired optimization to

<sup>&</sup>lt;sup>1</sup>The project CG Learning acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 255827.

<sup>&</sup>lt;sup>2</sup>Institute of Theoretical Computer Science, ETH Zürich, stich@inf.ethz.ch

<sup>&</sup>lt;sup>4</sup>Institute of Theoretical Computer Science, ETH Zürich, cm192@nyu.edu

 $<sup>^3</sup> Institute of Theoretical Computer Science, ETH Zürich, {\tt gaertner@inf.ethz.ch}$ 

mathematical programming. Important examples include the Gaussian Adaptation algorithm developed by Kjellström and Taxen [16, 20] in the context of analog circuit design, Marti's controlled random search schemes using concepts from optimal control [18], and the arguably most popular scheme, Hansen's Evolution Strategy with Covariance Matrix Adaptation (CMA-ES) [10] that emerged in the bio-inspired optimization community.

Despite their great appeal in practice many randomized gradient-free variable metric schemes lack a thorough theoretical convergence analysis. A marked exception is Leventhal and Lewis' recent work on Randomized Hessian approximation (RHE) [17]. We here adopt some of their ideas and extend our framework of Random Pursuit (RP) [32], eventually leading to Variable Metric Random Pursuit (V-RP) schemes. We solely consider optimization problems of the kind:

min 
$$f(\mathbf{x})$$
 subject to  $\mathbf{x} \in \mathbb{R}^n$ , (1)

where f is a smooth convex function. We assume that there is a global minimum and that the curvature of the function f is bounded from above. Moreover, we assume that we have only access to function values of f. No analytic gradient or higher order information about f is available.

To motivate Variable Metric Random Pursuit, let us first sketch the working mechanism of standard Random Pursuit on an illustrative example. Each iteration of standard Random Pursuit consists of two steps: (i) a random direction is sampled from an isotropic probability distribution; (ii) the next iterate is chosen such as to (approximately) minimize the objective function along this direction. In [32] we have shown that the expected error in function value decreases by a factor of  $(1 - \frac{m}{n\ell})$  in every step, if m > 0 and  $\ell > 0$  are parameters of quadratic functions that bound the difference between f and any of its linear approximations from below and above<sup>1</sup>; more precisely,

$$\frac{m}{2} \left\| \mathbf{y} - \mathbf{x} \right\|^2 \le \ell_{\mathbf{x}}(\mathbf{y}) := f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle \le \frac{\ell}{2} \left\| \mathbf{y} - \mathbf{x} \right\|^2$$
(2)

is assumed to hold for all  $\mathbf{x}, \mathbf{y}$ . For twice differentiable functions this condition is equivalent to an uniform lower and upper bound on the Hessian  $H(\mathbf{x})$ :  $m \leq H(\mathbf{x}) \leq \ell$ . As an example, let us consider the function

$$f_0(x_1, x_2) = 100x_1^2 + x_2^2,$$

for which  $\ell_{\mathbf{x}}(\mathbf{y}) = 100(x_1 - y_1)^2 + (x_2 - y_2)^2$ . This means that m = 2 and  $\ell = 200$  are the best possible parameters in (2), and the progress rate in every step is no better than (1 - 1/200). This also matches our intuition: every level set of  $f_0$  is a long and skinny ellipse, stretching out along the  $x_2$ -axis; if we start from a point close to the  $x_2$  axis, the progress in a step will be small, unless we almost sample in  $x_2$ -direction.

For this particular function  $f_0$ , it would be better to sample from an anisotropic distribution that favors the  $x_2$ -direction. Once we fix such an anisotropic sampling distribution, however, other functions become "bad"; in fact, without prior knowledge about f, anisotropic sampling makes no sense at all. Here is where the "variable metric" approach comes in. The idea is to gradually *adapt* the

 $<sup>^1{\</sup>rm The}$  left inequality is usually referred to as strong-convexity; the right one follows from Lipschitz continuity of the gradient. See Section 3.

sampling distribution to the function f while we run the algorithm. Suppose that we can somehow estimate the Hessians at the various iterates. Under the assumption that f is wedged between two quadratic functions—whose Hessians are not necessarily multiples of the identity, as in (2)—these estimates will allow us to learn a suitable metric that guides the sampling distribution. In case of  $f_0$ , we would start with the isotropic one and then converge to a distribution that indeed favors the  $x_2$ -direction with the right proportion.

In this contribution we present a framework for analyzing the convergence behavior of Random Pursuit algorithms on convex functions. In a first step we analyze the Fixed Metric Random Pursuit (F-RP) algorithm for fixed (anisotropic) sampling distributions. In a second step we equip Random Pursuit with a randomized scheme to update the metric that defines the sampling distribution in every step: the Variable Metric Random Pursuit. We present precise theoretical analysis of an update scheme recently proposed by Leventhal and Lewis [17] as well as three novel implementations.. These learning schemes are generic in the sense that they work for all convex functions and do not require any prior knowledge of the function's shape. We prove that the sampling distribution converges to a distribution that yields asymptotically optimal (and function-independent) progress rates. The proposed schemes are easily parallelizable, thus allowing a computational speed-up of the update schemes on multi-core machines.

The remainder of the paper is structured as follows. In Section 2 we give a generic description of the different Random Pursuit algorithms and their essential building blocks. We introduce all relevant mathematical definitions such as matrix upper and lower bounds of convex functions and expressions for certain scalar and matrix expectations in Section 3. We derive the expected single-step progress and global convergence of F-RP in Section 4. Section 5 is dedicated to Variable Metric Random Pursuit. We discuss the key results of the paper and outline future research goals in Section 6.

# 2 Fixed and Variable Metric Random Pursuit

All Random Pursuit algorithms are designed for problems as in (1). Before stating the formal definition of the considered RP algorithms we need to define one indispensable primitive.

**Definition 1** (Line search oracle). For  $\mathbf{x} \in \mathbb{R}^n$ , a direction  $\mathbf{u} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$  and a convex function f, a function  $LS_f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$  with

$$LS_f(\mathbf{x}, \mathbf{u}) = \operatorname*{arg\,min}_{h \in \mathbb{R}} f(\mathbf{x} + h\mathbf{u})$$
(3)

is called an exact line search oracle.

The two RP schemes considered here are summarized in Fig. 1. In Fixed Metric Random Pursuit (F-RP) a direction  $\mathbf{u} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$  is sampled from a multivariate normal distribution with fixed covariance  $\Sigma$  at iteration k of the algorithm. The next iterate  $\mathbf{x}_k$  is calculated from the current iterate  $\mathbf{x}_{k-1}$  as

$$\mathbf{x}_k := \mathbf{x}_{k-1} + \mathsf{LS}_f(\mathbf{x}_{k-1}, \mathbf{u}) \cdot \mathbf{u} \,. \tag{4}$$

This algorithm only requires function evaluations in addition to the line search oracle. No first or second-order information about the objective is needed. We emphasize that besides the starting point no further input parameters describing function properties (such as curvature constant etc.) are necessary. The actual run time will, however, depend on the specific properties of the objective function and on the choice of the covariance matrix  $\Sigma$ , as detailed in Section 4. Variable Metric Random Pursuit (V-RP) comprises an independent process that gives an approximation of the Hessian at each iteration. The inverse of the Hessian is then used as covariance matrix in the multivariate normal distribution to generate the current search direction. In principle, any deterministic or randomized gradient-free estimator can be used for this purpose. In Section 5 we will use a Randomized Hessian approximation scheme recently proposed in [17] for this task.

For simplicity we assumed here access to an exact line search oracle. However, approximate line search schemes are sufficient to establish convergence of the Random Pursuit algorithms. We introduce such oracles in Section 3.4.

# 3 Definitions and Notations

We now introduce the notation and some inequalities that will be useful for the subsequent analysis. Most importantly, we define two classes of convex functions with respect to so-called quadratic norms. This extends the standard model and allows us to derive convergence rates that take the eigenvalue spectrum of the Hessian into account.

### 3.1 Quadratic norms

Let  $\mathrm{PD}_n$  denote the set of symmetric positive definite  $n \times n$  matrices. With respect to  $A \in \mathrm{PD}_n$ , we can define an 'anisotropic' norm by  $\|\mathbf{x}\|_A^2 := \langle \mathbf{x}, \mathbf{x} \rangle_A$  for  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ . In statistics this metric is also known as the Mahalanobis metric. We observe that

$$\lambda_{\min}(A) \left\| \mathbf{x} \right\|^2 \le \left\| \mathbf{x} \right\|_A^2 \le \lambda_{\max}(A) \left\| \mathbf{x} \right\|^2, \tag{5}$$

due to  $\lambda_{\min}(A) = \min\{\mathbf{x}^T A \mathbf{x} : ||x|| = 1\}$  and  $\lambda_{\max}(A) = \max\{\mathbf{x}^T A \mathbf{x} : ||x|| = 1\}$ . This statement can be generalized, as shown in the following lemma.

$F-RP(f, \mathbf{x}_0, \Sigma, N)$	$\texttt{V-RP}(f,\mathbf{x}_0,B_0,N)$
<b>Output</b> : Approximate solution $x_N$	<b>Output</b> : Approximate solution $x_N$
to $(1)$	to $(1)$
1 for $k = 1$ to N do	1 for $k = 1$ to N do
2 $  \mathbf{u}_k \sim \mathcal{N}(0, \Sigma)$	$2 \mid B_k \leftarrow \texttt{updateHess}(f, \mathbf{x}, B_{k-1})$
$\mathbf{s} \ \ \mathbf{x}_k \leftarrow \mathtt{LS}_f(\mathbf{x}_{k-1}, \mathbf{u}_k)$	$\mathbf{a} \mid \mathbf{u}_k \sim \mathcal{N}(0, B_k^{-1})$
4 return $\mathbf{x}_N$	$4  \left\lfloor \mathbf{x}_k \leftarrow \mathtt{LS}_f(\mathbf{x}_{k-1}, \mathbf{u}_k) \right.$
	5 return $\mathbf{x}_N$

Figure 1: Fixed Metric Random Pursuit (left panel) and the Variable Metric version (right panel). The generic sub-routine updateHess on line 2 exemplifies any function that generates the metric  $B_k$  in step k. Three specific instantiations are discussed in Sec. 5 (cf. Fig. 3).

**Lemma 1.** Let  $A, B \in PD_n$  and  $\mathbf{x} \in \mathbb{R}^n$ . Then

$$\lambda_{\min}(B^{-1}A) \|\mathbf{x}\|_{B}^{2} \le \|\mathbf{x}\|_{A}^{2} \le \lambda_{\max}(B^{-1}A) \|\mathbf{x}\|_{B}^{2} .$$
 (6)

A proof can be found in [25, Prop. 18.3]. The substitution  $\mathbf{x} = (B^{1/2})^{-1}\mathbf{y}$ , where  $B^{1/2} \in \text{PD}_n$  denotes the positive semidefinite root of B and  $\mathbf{y} \in \mathbb{R}^n$ , allows to reduce (6) to (5). It remains to note that  $(B^{1/2})^{-1}A(B^{1/2})^{-1}$  and  $AB^{-1}$  have the same eigenvalues, for this see e.g. again [25, Prop. 13.2].

### 3.2 Quadratic bounds

We now define two function classes. We assume that the objective function f in (1) is differentiable and convex. The latter property is equivalent to

$$f(\mathbf{y}) \ge f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^n.$$
 (7)

We also require that the curvature of f is bounded. However, we allow for different curvatures depending on the direction. By this we mean that for some fixed symmetric and positive definite matrix  $L \in PD_n$ ,

$$f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle \le \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_{L}^{2}, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^{n}.$$
 (8)

We will also refer to this inequality as the *(matrix) quadratic upper bound*. We denote by  $C_L^1$  the class of (once) differentiable convex functions for which (8) holds with parameter L. A differentiable function is *strongly convex* with parameter  $M \in PD_n$  if the *(matrix) quadratic lower bound* 

$$f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle \ge \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_M^2 , \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^n ,$$
(9)

holds. Let  $\mathbf{x}^*$  be the unique minimizer of a strongly convex function f with parameter M. Then equation (9) implies this useful relation:

$$\frac{1}{2} \|\mathbf{x} - \mathbf{x}^*\|_M^2 \le f(\mathbf{x}) - f(\mathbf{x}^*) \le \frac{1}{2} \|\nabla f(\mathbf{x})\|_{M^{-1}}^2, \quad \forall \mathbf{x} \in \mathbb{R}^n.$$
(10)

The former inequality uses  $\nabla f(\mathbf{x}^*) = 0$ , and the latter one follows from (9) via

$$f(\mathbf{x}^*) \ge f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{x}^* - \mathbf{x} \rangle + \frac{1}{2} \| \mathbf{x}^* - \mathbf{x} \|_M^2$$
  
$$\ge f(\mathbf{x}) + \min_{\mathbf{y} \in \mathbb{R}^n} \left( \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{1}{2} \| \mathbf{y} - \mathbf{x} \|_M^2 \right) = f(\mathbf{x}) - \frac{1}{2} \| \nabla f(\mathbf{x}) \|_{M^{-1}}^2$$

by standard calculus.

### 3.3 Sampling distribution

Both RP algorithms from Figure 1 rely on multivariate normal distributed search directions  $\mathbf{u} \in \mathbb{R}^n$ . We write  $\mathbf{u} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$  to denote that  $\mathbf{u} \in \mathbb{R}^n$  is multivariate normally distributed with mean  $\boldsymbol{\mu} \in \mathbb{R}^n$  and covariance  $\Sigma \in PD_n$ . As the step sizes in (4) are determined by a line search, the actual scaling of  $\mathbf{u}$ , i.e.  $\|\mathbf{u}\|$ , is not relevant for the behavior of the algorithm. We therefore restrict ourselves to *normalized* search directions. **Definition 2** (Normalized distribution). Let  $\Sigma \in PD_n$ . We denote by  $\overline{\mathcal{N}}(\mathbf{0}, \Sigma)$  the distribution arising from the image of the normal distribution  $\mathcal{N}(\mathbf{0}, \Sigma)$  under the mapping  $T(\mathbf{x}) = \mathbf{x}/\|\mathbf{x}\|_{\Sigma^{-1}}$ .

For example,  $\mathbf{u} \sim \overline{\mathcal{N}}(\mathbf{0}, I_n)$  denotes the uniform distribution over all unit length vectors subject to the standard Euclidean norm (the uniform distribution on the unit (n-1)-sphere). The following lemma summarizes some facts for the normalized distribution.

**Lemma 2.** Let  $\mathbf{v} \sim \overline{\mathcal{N}}(\mathbf{0}, \Sigma)$  normalized with  $\Sigma \in PD_n$  and let  $A \in SYM_n$ . Then

$$\mathbb{E}\left[\mathbf{v}\mathbf{v}^{T}\right] = \frac{\Sigma}{n}, \quad \mathbb{E}\left[\mathbf{v}^{T}A\mathbf{v}\right] = \frac{\operatorname{Tr}[A\Sigma]}{n}, \quad \mathbb{E}\left[(\mathbf{v}^{T}A\mathbf{v})^{2}\right] = \frac{\operatorname{Tr}[A\Sigma]^{2} + 2\operatorname{Tr}[(A\Sigma)^{2}]}{n(n+2)}$$

and for  $\mathbf{x} \in \mathbb{R}^n$ ,

$$\mathbb{E}\left[\langle \mathbf{x}, \mathbf{v} \rangle \mathbf{v}\right] = \frac{\Sigma \mathbf{x}}{n}, \quad and \quad \mathbb{E}\left[\left\|\langle \mathbf{x}, \mathbf{v} \rangle \mathbf{v}\right\|_{A}^{2}\right] = \frac{\operatorname{Tr}[A\Sigma] \left\|\mathbf{x}\right\|_{\Sigma}^{2} + 2\left\|\mathbf{x}\right\|_{\Sigma A \Sigma}^{2}}{n(n+2)}.$$

The proof can be found on page 28 in the appendix.

# 3.4 Approximate line search oracles

Access to an exact line search oracle (3) is typically not required to establish convergence of the RP algorithms. This is of importance in practical applications. Commonly used line search oracles often aim at satisfying the well-known Armijo-Goldstein [9, 3], and Wolfe [35, 36] conditions. These condition measure the quality of a single search step in terms of the squared norm of the gradient. Thus, we also provide an analogous quality criterion in the full quadratic model—in addition to a slightly stronger relative accuracy measure.

**Definition 3** (Approximate line search oracles). For  $0 \leq \mu \leq 1$ ,  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{u} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$  and a convex function f, a function  $ALS_f \colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$  with

$$f(\mathbf{x} + ALS_f(\mathbf{x}, \mathbf{u}) \cdot \mathbf{u}) \le f(\mathbf{x}) - \mu(f(\mathbf{x}) - f(\mathbf{x} + LS_f(\mathbf{x}, \mathbf{u}) \cdot \mathbf{u}))$$
(L1)

is called an approximate line search oracle with relative accuracy  $\mu$ .

For a differentiable convex function  $f \in C_L^1$ , a function  $ALS_f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ with

$$f(\mathbf{x} + \mathsf{ALS}_f(\mathbf{x}, \mathbf{u}) \cdot \mathbf{u}) \le f(\mathbf{x}) - \frac{\mu \langle \nabla f(\mathbf{x}, \mathbf{u})^2}{2 \|\mathbf{u}\|_L^2}$$
(L2)

is called an approximate line search oracle with sufficient decrease  $\mu$ .

As we measure deviations only on a relative and not on an absolute scale, such an inexact line search oracle can efficiently be implemented with binary search (dichotomy), using function evaluations only. It can easily be seen that the first condition (L1) is stronger than (L2).

**Lemma 3** (L1)  $\Rightarrow$  (L2). Let  $\mathbf{x} \in \mathbb{R}^n$ , function  $f \in C_L^1$ , search direction  $\mathbf{u} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$  and  $\operatorname{ALS}_f$  a line search oracle with relative accuracy  $\mu > 0$ . Then  $\operatorname{ALS}_f$  satisfies the sufficient decrease condition (L2).

*Proof.* As a simple consequence of (3) we have  $f(\mathbf{x} + \mathsf{LS}_f(\mathbf{x}, \mathbf{u}) \cdot \mathbf{u}) \leq f(\mathbf{x} + t\mathbf{u})$  for every  $t \in \mathbb{R}$ . We use the quadratic upper bound (8) to derive an upper bound on  $f(\mathbf{x} + t\mathbf{u})$ . Assume  $\mu = 1$ . Therefore

$$f(\mathbf{x} + \mathsf{LS}_f(\mathbf{x}, \mathbf{u})\mathbf{u}) \le f(\mathbf{x}) + \min_{t \in \mathbb{R}} \left( t \left\langle \nabla f(\mathbf{x}), \mathbf{u} \right\rangle + t^2 \frac{1}{2} \|\mathbf{u}\|_L^2 \right)$$
(11)

And the lemma follows by the (now optimal) choice  $t = -\frac{\langle \nabla f(\mathbf{x}), \mathbf{u} \rangle}{2 \|\mathbf{u}\|_{L}^{2}}$ . The general case  $\mu < 1$  follows straightforwardly from definition (L2).

Most of our convergence results hold for both approximate line search oracles, but Theorem 3 will rely on the stronger oracle (L1). We would like to remark that our convergence results to more general settings. For instance, we can vary the accuracy parameter  $\mu$  in every iteration as long as  $\mu$  stays positive, or the distribution of  $\mu$  is independent of  $\mathbf{x}$  and  $\mathbf{u}$  (see also the concrete implementations in Section 5.4).

So far, we did not discuss line search oracles with *absolute* errors. We will comment on such oracles in Section 4.3 below.

#### 3.5 Convergence factors

The following notation will be useful to formulate the convergence results form Section 4 below. The condition number  $\kappa(A)$  of a positive definite matrix  $A \in$  $\mathrm{PD}_n$  is defined as the ratio  $\kappa(A) := \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$  of the two most extreme eigenvalues. The quantities that we introduce now, can be viewed as a generalization of this concept. For  $A, B, C, D \in \mathrm{PD}_n$ , and  $\mathbf{y} \in \mathbb{R}^n$  let

$$\kappa_{\rm E}(A, B, C, \mathbf{y}) := \frac{\operatorname{Tr}[AB]\sigma_{A,B}(\mathbf{y}) + 2}{\lambda_{\min}(C)(n+2)}, \quad \sigma_{A,B}(\mathbf{y}) := \frac{\|\mathbf{y}\|_{(ABA)^{-1}}^2}{\|\mathbf{y}\|_{A^{-1}}^2}, \quad (12)$$

and

$$\kappa_{\mathrm{T}}(D,C) := \frac{\mathrm{Tr}[D]\lambda_{\min}^{-1}(D) + 2}{\lambda_{\min}(C)(n+2)}.$$
(13)

For brevity, we abbreviate  $\kappa_{\mathrm{T}}(D) := \kappa_{\mathrm{T}}(D, I_n)$ .

**Lemma 4.** Let  $A, B, C \in PD_n$ , and  $\mathbf{y} \in \mathbb{R}^n$ . Then

$$0 < \kappa_{\mathrm{E}}(A, B, C, \mathbf{y}) \le \kappa_{\mathrm{T}}(AB, C) \le \frac{\frac{1}{n} \mathrm{Tr}[AB] \lambda_{\min}^{-1}(AB)}{\lambda_{\min}(C)} \le \frac{\kappa(AB)}{\lambda_{\min}(C)}$$

*Proof.* We show the inequalities one by one. For the first one it is enough to show that Tr[AB] is positive. Let  $A^{1/2} \in \text{PD}_n$  denote the positive definite root of A. Then AB and  $A^{1/2}BA^{1/2} \in \text{PD}_n$  have the same eigenvalues, as already mentioned in Section 3.1 above, see e.g. [25, Prop. 13.2]. For the second one we use Lemma 1 to find a uniform upper bound on  $\sigma_{A,B}(\mathbf{y})$ :

$$\sigma_{A,B}(\mathbf{y}) = \frac{\|\mathbf{y}\|_{(ABA)^{-1}}^2}{\|\mathbf{y}\|_{A^{-1}}^2} \le \lambda_{\max}(A^{-1}B^{-1}) = \lambda_{\min}^{-1}(AB).$$
(14)

For  $a \ge b > 0$  it holds  $\frac{a+c}{b+c} \le \frac{a}{b}$  for any  $c \ge 0$ . Therefore, the choice  $a = \text{Tr}[AB]\lambda_{\min}^{-1}(AB)$ , b = n and c = 2 implies the third inequality. The last one is trivial.

# 4 Convergence of Fixed Metric Random Pursuit

We will now derive the global convergence rates for Algorithm F-RP on convex and strongly convex functions. To prepare the proof, we first study the expected progress in a single step, which is the quantity

$$f(\mathbf{x}_k) - \mathbb{E}\left[f(\mathbf{x}_{k+1}) \mid \mathbf{x}_k\right]$$

For the first two theorems, it suffices to assume access to an approximate line search oracle with (L2). However, for Theorem 3, the stronger (L1) is required.

# 4.1 Single step progress

Once a search direction is determined, the subsequent iterate is chosen according to (4). As the step size is determined by the line search oracle, we can derive the following lower bound on the single step progress.

**Lemma 5** (Single step progress of (L2)). Let  $f \in C_L^1$ ,  $\mathbf{x} \in \mathbb{R}^n$  such that  $\nabla f(\mathbf{x}) \neq \mathbf{0}$ , covariance  $\Sigma \in \text{PD}_n$  direction  $\mathbf{u} \sim \overline{\mathcal{N}}(\mathbf{0}, \Sigma)$ , and ALS an approximate line search oracle (L2) with sufficient decrease  $0 \leq \mu \leq 1$  and let  $\mathbf{x}_+ = \mathbf{x} + \text{ALS}_f(\mathbf{x}, \mathbf{u}) \cdot \mathbf{u}$  the next iterate after one step of Algorithm F-RP. Then

$$\begin{split} \mathbb{E}_{\mathbf{u}}\left[f(\mathbf{x}_{+}) \mid \mathbf{x}\right] &= f(\mathbf{x}) - \frac{\mu}{2n\kappa_{\mathrm{E}}(L,\Sigma,I_{n},\nabla f(\mathbf{x}))} \left\|\nabla f(\mathbf{x})\right\|_{L^{-1}}^{2} \\ &\leq f(\mathbf{x}) - \frac{\mu}{2n\kappa_{\mathrm{T}}(L\Sigma,I_{n})} \left\|\nabla f(\mathbf{x})\right\|_{L^{-1}}^{2} \,. \end{split}$$

where  $\kappa_{\rm E}$  and  $\kappa_{\rm T}$  as in Section 3.5.

Proof. All we need to find is a lower bound on the conditional expectation

$$E_L := \mathbb{E}\left[\frac{\langle \nabla f(\mathbf{x}), \mathbf{u} \rangle^2}{2 \|\mathbf{u}\|_L^2} \mid \mathbf{x}\right], \qquad (15)$$

of the expression on the right hand side of (L2). Expressions for such expected values have been derived in the literature (see e.g. [19]), but no simple closed form solutions exist. As we here only need a lower bound, we can apply the following trick. For fixed  $\mathbf{y} \in \mathbb{R}^n$  and  $\mathbf{u} \in \mathbb{R}^n \setminus \{0\}$  we observe

$$\begin{aligned} \frac{\left\langle \mathbf{y}, \mathbf{u} \right\rangle^2}{\left\| \mathbf{u} \right\|_L^2} &= \max_t \left( 2t \left\langle \mathbf{y}, \mathbf{u} \right\rangle - t^2 \left\| \mathbf{u} \right\|_L^2 \right) \\ &\geq \max_h \left( 2h \left\langle (L\Sigma)^{-1} \mathbf{y}, \mathbf{u} \right\rangle \left\langle \mathbf{y}, \mathbf{u} \right\rangle - h^2 \left\| \left\langle (L\Sigma)^{-1} \mathbf{y}, \mathbf{u} \right\rangle \mathbf{u} \right\|_L^2 \right) ,\end{aligned}$$

where the equality follows by standard calculus, and the inequality by suboptimally setting  $t = h \langle (L\Sigma)^{-1} \mathbf{y}, \mathbf{u} \rangle$ . With Lemma 2 we can compute the expectation of the terms inside the maximum. We have

$$\mathbb{E}_{\mathbf{u}}\left[\left\langle (L\Sigma)^{-1}\mathbf{y},\mathbf{u}\right\rangle \langle \mathbf{y},\mathbf{u}\rangle \mid \mathbf{x}\right] = \mathbb{E}_{\mathbf{u}}\left[\mathbf{u}^{T}(L\Sigma)^{-1}\mathbf{y}\mathbf{y}^{T}\mathbf{u} \mid \mathbf{x}\right] = \frac{1}{n}\left\|\mathbf{y}\right\|_{L^{-1}}^{2},$$

and

$$\mathbb{E}_{\mathbf{u}}\left[\left\|\left\langle (L\Sigma)^{-1}\mathbf{y},\mathbf{u}\right\rangle\mathbf{u}\right\|_{L}^{2} \mid \mathbf{x}\right] = \frac{\operatorname{Tr}[L\Sigma] \left\|\mathbf{y}\right\|_{(L\Sigma L)^{-1}}^{2} + 2\left\|\mathbf{y}\right\|_{L^{-1}}^{2}}{n(n+2)}.$$

By Jensen's inequality it is indeed valid to interchange the expectation with the maximum. We have

$$E_{L} \geq \max_{h} \left( 2h \frac{\|\mathbf{y}\|_{L^{-1}}^{2}}{n} - h^{2} \frac{\operatorname{Tr}[L\Sigma] \|\mathbf{y}\|_{(L\Sigma L)^{-1}}^{2} + 2 \|\mathbf{y}\|_{L^{-1}}^{2}}{n(n+2)} \right)$$
$$\geq \frac{(n+2) \|\mathbf{y}\|_{L^{-1}}^{4}}{n(\operatorname{Tr}[L\Sigma] \|\mathbf{y}\|_{(L\Sigma L)^{-1}}^{2} + 2 \|\mathbf{y}\|_{L^{-1}}^{2})},$$

where h was chosen to maximize the expression in the bracket, i.e.

$$-(n+2) \|\mathbf{y}\|_{L^{-1}}^{2} + h\left(\operatorname{Tr}[L\Sigma] \|\mathbf{y}\|_{(L\Sigma L)^{-1}}^{2} + 2 \|\mathbf{y}\|_{L^{-1}}^{2}\right) = 0.$$

This choice of h implies the first inequality for  $\mathbf{y} = \nabla f(\mathbf{x})$ . The second one follows directly from Lemma 4.

The line search oracle with absolute accuracy (L1) achieves a single step progress that is as least as good as the bound derived in Lemma 5 above. However, we are more flexible and an can also derive a bound that does not scale directly with  $\|\nabla f(\mathbf{x})\|_{L^{-1}}^2$ .

**Lemma 6** (Single step progress of ((L1)). Let  $f \in C_L^1$ ,  $\mathbf{x} \in \mathbb{R}^n$ , covariance  $\Sigma \in \text{PD}_n$  direction  $\mathbf{u} \sim \overline{\mathcal{N}}(\mathbf{0}, \Sigma)$ , and ALS an approximate line search oracle (L1) with relative accuracy  $0 \leq \mu \leq 1$  and let  $\mathbf{x}_+ = \mathbf{x} + \text{ALS}_f(\mathbf{x}, \mathbf{u}) \cdot \mathbf{u}$  the next iterate after one step of Algorithm F-RP. In addition, let  $\mathbf{x}^* \in \mathbb{R}^n$  be one of the minimizers of f. Then for every positive  $h \geq 0$  it holds

$$\mathbb{E}_{\mathbf{u}}\left[f(\mathbf{x}_{+}) - f(\mathbf{x}^{*}) \mid \mathbf{x}\right] \leq \left(1 - \frac{h\mu}{n}\right)\left(f(\mathbf{x}) - f(\mathbf{x}^{*})\right) + \frac{h^{2}\mu\kappa_{\mathrm{T}}(L\Sigma)}{2n} \left\|\mathbf{x} - \mathbf{x}^{*}\right\|_{L}^{2},$$

where  $\kappa_{\rm T}$  as in Section 3.5.

*Proof.* As in the proof of Lemma 3 we use a supoblimal choice of the unknown optimal value  $LS_f(\mathbf{x}, \mathbf{u})$  together with the quadratic upper bound. Here we use in (11) the value  $t = h \langle \Sigma^{-1}(\mathbf{x} - \mathbf{x}^*), \mathbf{u} \rangle$ . This leads to

$$f(\mathbf{x}_{+}) \leq f(\mathbf{x}) - h\mu \left\langle \Sigma^{-1}(\mathbf{x} - \mathbf{x}^{*}), \mathbf{u} \right\rangle \left\langle \nabla f(\mathbf{x}), \mathbf{u} \right\rangle + \frac{h^{2}\mu}{2} \left\| \left\langle \Sigma^{-1}(\mathbf{x} - \mathbf{x}^{*}), \mathbf{u} \right\rangle \cdot \mathbf{u} \right\|_{L}^{2}$$

With Lemma 2 we can again compute the conditional expectation of the terms on the right hand side:

$$\mathbb{E}_{\mathbf{u}}[\left\langle \Sigma^{-1}(\mathbf{x} - \mathbf{x}^*), \mathbf{u} \right\rangle \mathbf{u} \mid \mathbf{x}] = \frac{1}{n} (\mathbf{x} - \mathbf{x}^*) ,$$
$$\mathbb{E}_{\mathbf{u}}\left[\left\|\left\langle \Sigma^{-1}(\mathbf{x} - \mathbf{x}^*), \mathbf{u} \right\rangle \cdot \mathbf{u}\right\|_{L}^{2} \mid \mathbf{x}\right] = \frac{\operatorname{Tr}[L\Sigma] \left\|\mathbf{x} - \mathbf{x}^*\right\|_{\Sigma^{-1}}^{2} + 2 \left\|\mathbf{x} - \mathbf{x}^*\right\|_{L}^{2}}{n(n+2)} ,$$

and obtain

$$\mathbb{E}_{\mathbf{u}}[f(\mathbf{x}_{+}) \mid \mathbf{x}] \leq f(\mathbf{x}) - \frac{h\mu}{n} \langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{x}^{*} \rangle + \frac{h^{2}\mu}{2n(n+2)} \left( \operatorname{Tr}[L\Sigma] \|\mathbf{x} - \mathbf{x}^{*}\|_{\Sigma^{-1}}^{2} + 2 \|\mathbf{x} - \mathbf{x}^{*}\|_{L}^{2} \right).$$
(16)

Using the definition of convexity (see the beginning of Section 3.2) we can bound the term  $\langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{x}^* \rangle$  from below by  $f(\mathbf{x}) - f(\mathbf{x}^*)$ . Finally, we bound the  $\Sigma^{-1}$ -norm from above with Lemma 1. As in the proof of Lemma 4 we get

$$\|\mathbf{x} - \mathbf{x}^*\|_{\Sigma^{-1}}^2 \le \lambda_{\max}(L^{-1}\Sigma^{-1}) \|\mathbf{x} - \mathbf{x}^*\|_L$$

and the lemma follows from  $\lambda_{\max}(L^{-1}\Sigma^{-1}) = 1/\lambda_{\min}(\Sigma L)$ .

The previous two lemmas shows that, on average, there is progress in every single step if either  $\|\nabla f(\mathbf{x})\|_{L^{-1}}$  or  $\|\mathbf{x} - \mathbf{x}^*\|_L^2$  is bounded away from zero.<sup>2</sup> This leads us to the next section where we will use the just derived lemmas to prove global convergence.

# 4.2 Global convergence

We now use the previously derived bounds on the expected single step progress (Lemma 5 and 6) to show convergence of F-RP in expectation. We first show convergence on smooth but not necessarily strongly convex functions.

**Theorem 1.** Let  $f \in C_L^1$ , let  $\mathbf{x}^* \in \mathbb{R}^n$  be a minimizer of f and let the sequence  $\{\mathbf{x}_k\}_{k\geq 0}$  be generated by Algorithm F-RP with covariance  $\Sigma \in \text{PD}_n$  and line search (L2) with sufficient decrease  $0 < \mu \leq 1$ . Assume there exists  $R \in \mathbb{R}$ , s.t.  $\|\mathbf{y} - \mathbf{x}_0\|_L \leq R$  for all  $\mathbf{y} \in \mathbb{R}^n$  with  $f(\mathbf{y}) \leq f(\mathbf{x}_0)$ . Then, for any  $N \geq 0$ , we have

$$\mathbb{E}\left[f(\mathbf{x}_N) - f(\mathbf{x}^*)\right] \le \frac{Q}{N+1}\,,$$

where

$$Q := \max\left\{\frac{2nR^2\kappa_{\mathrm{T}}(L\Sigma)}{\mu}, f(\mathbf{x}_0) - f(\mathbf{x}^*)\right\}.$$

*Proof.* We will apply the bound on the single step progress from Lemma 5, but first, let us derive a lower bound on the norm  $\|\nabla f(\mathbf{x})\|_{L^{-1}}^2$  for  $\mathbf{x} \in \mathbb{R}^n$  with  $\|\mathbf{x} - \mathbf{x}^*\|_L \leq R$ . By convexity (7) and the assumptions on  $\mathbf{x}$  we have

$$f(\mathbf{x}) - f(\mathbf{x}^*) \le \langle \nabla f(\mathbf{x}), \mathbf{x}^* - \mathbf{x} \rangle \le R \| \nabla f(\mathbf{x}) \|_{L^{-1}}$$

Hence, by Lemma 5 we can estimate the single step progress

$$\mathbb{E}\left[f(\mathbf{x}_{+}) \mid \mathbf{x}\right] \leq f(\mathbf{x}) - \tau \left(f(\mathbf{x}) - f(\mathbf{x}^{*})\right)^{2}$$

where  $\tau := \frac{\mu}{2nR^2\kappa_{\mathrm{T}}(L\Sigma)}$  and  $\mathbf{x}_+ = \mathbf{x} + \mathrm{ALS}_f(\mathbf{x}, \mathbf{u}) \cdot \mathbf{u}$  with the notation from Lemma 5. Conditioned on  $\mathbf{x}$ , the quantity  $f(\mathbf{x}) - f(\mathbf{x}^*) =: f_{\mathbf{x}}$  is just a constant. Hence, subtracting  $f(\mathbf{x}^*)$  on both sides, we can rewrite this bound as

$$\mathbb{E}\left[f(\mathbf{x}_{+}) - f(\mathbf{x}^{*}) \mid \mathbf{x}\right] \leq f_{\mathbf{x}} - \tau f_{\mathbf{x}}^{2} = f_{\mathbf{x}} + 2\min_{h} \left(-hf_{\mathbf{x}} + \frac{h^{2}}{2\tau}\right)$$
$$\leq (1 - 2h)f_{\mathbf{x}} + h^{2}\tau^{-1}, \qquad (17)$$

where the last inequality holds for arbitrary parameter  $h \in \mathbb{R}$ .

<sup>&</sup>lt;sup>2</sup>Here we use that  $\text{Tr}[L\Sigma] > 0$ ; see the proof of Lemma 4 in Section 3.5.

Now we can proceed to analyze the multi step behavior. For this, we just repeatedly apply the bound (17) on the single step progress. Conditioning on  $\{\mathbf{x}_k\}_{k=0}^{N-1}$ , we estimate

$$\mathbb{E}\left[f(\mathbf{x}_N) - f(\mathbf{x}^*) \mid \{\mathbf{x}_k\}_{k=0}^{N-1}\right] \le (1 - 2h_N)f_{\mathbf{x}} + \frac{h_N^2}{\tau}$$

for any parameter  $h_N \in \mathbb{R}$ . Now, formally, we recursively apply the conditional expectations, onditioning on  $\{\mathbf{x}_k\}_{k=0}^{N-2}$ ,  $\{\mathbf{x}_k\}_{k=0}^{N-3}$ , ...,  $\{\mathbf{x}_0\}$ , and use (17) with different parameters  $h_{N-1}, \ldots, h_1$  in every step. By the tower property of conditional expectations, we end up with a bound on  $\mathbb{E}[f(\mathbf{x}_N) - f(\mathbf{x}^*)]$  that depends on the free parameters  $h_1, \ldots, h_N$ . As in [32, Theorem 5.3], the choice  $h_k := \frac{1}{k}$  for  $k = 1, \ldots, N$  yields the lemma (see also [32, Lemma A.1]).

On strongly convex functions the convergence of F-RP is linear.

**Theorem 2.** Let  $f \in C_L^1$  and let f in addition be strongly convex with parameter  $M \in \text{PD}_n$ . Let  $\mathbf{x}^* \in \mathbb{R}^n$  denote the unique minimizer of f, and let the sequence  $\{\mathbf{x}_k\}_{k\geq 0}$  be generated by Algorithm F-RP with covariance  $\Sigma \in \text{PD}_n$  and line search with accuracy  $0 \leq \mu \leq 1$ . Then

$$\mathbb{E}\left[f(\mathbf{x}_N) - f(\mathbf{x}^*)\right] \le \left(1 - \frac{\mu}{n\kappa_{\mathrm{T}}(L\Sigma, M)}\right)^N \cdot \left(f(\mathbf{x}_0) - f(\mathbf{x}^*)\right) \,.$$

Proof. We use Lemma 1 to establish

$$\|\nabla f(\mathbf{x}_k)\|_{L^{-1}}^2 \ge \lambda_{\min}(ML^{-1}) \|\nabla f(\mathbf{x}_k)\|_{M^{-1}}^2$$

Applying the quadratic lower bound (10) to further bound the latter term from below yields

$$\|\nabla f(\mathbf{x}_k)\|_{L^{-1}}^2 \ge 2\lambda_{\min}(ML^{-1})(f(\mathbf{x}_k) - f(\mathbf{x}^*))$$
.

Now we can combine this bound with Lemma 5 and get

$$\mathbb{E}_{\mathbf{u}}\left[f(\mathbf{x}_k) - f(\mathbf{x}^*) \mid \mathbf{x}_k\right] \le \varrho(\mathbf{x}_k) \cdot \left(f(\mathbf{x}_k) - f(\mathbf{x}^*)\right), \tag{18}$$

where

$$\varrho(\mathbf{x}_k) := 1 - \frac{\mu}{n\kappa_{\rm E}(L, \Sigma, M, \nabla f(\mathbf{x}_k))} \le 1 - \frac{\mu}{n\kappa_{\rm T}(L\Sigma, M)}, \qquad (19)$$

is the exact convergence factor. The uniform upper bound was established in Lemma 4. The Theorem follows now by taking expectation over  $\mathbf{x}_k$ .  $\Box$ 

We remark that the progress is strict: by Lemma 4 the convergence factor

$$\hat{\varrho} := 1 - \frac{\mu}{n\kappa_{\mathrm{T}}(L\Sigma, M)}, \qquad (20)$$

is strictly smaller than one.

It is not necessary that the function f is strongly convex everywhere for linear convergence to hold. Theorem 3 below shows that convergence (at about a quarter of the rate of the one in Theorem 2) can be proven assuming only a weaker condition. Let us recall that strong convexity with parameter M implies that

$$f(\mathbf{x}) - f(\mathbf{x}^*) \ge \frac{1}{2} \|\mathbf{x} - \mathbf{x}^*\|_M^2, \forall \mathbf{x} \in \mathbb{R}^n.$$
(21)

It turns out that, instead of strong convexity (9), the weaker condition (21) is enough for linear convergence. Strong convex functions need to have positive curvature everywhere, whereas functions with (21) could also be linear on bounded subsets.

**Theorem 3.** Let  $f \in C_L^1$  and let f in addition have a unique minimizer  $\mathbf{x}^* \in \mathbb{R}^n$  satisfying (21) with  $M \in \text{PD}_n$ . Let the sequence  $\{\mathbf{x}_k\}_{k\geq 0}$  be generated by Algorithm F-RP with covariance  $\Sigma \in \text{PD}_n$  and line search oracle (L1) with relative accuracy  $0 \leq \mu \leq 1$ . Then

$$\mathbb{E}\left[f(\mathbf{x}_N) - f(\mathbf{x}^*)\right] \le \left(1 - \frac{\mu}{4n\kappa_{\mathrm{T}}(L\Sigma, ML^{-1})}\right)^N \cdot \left(f(\mathbf{x}_0) - f(\mathbf{x}^*)\right)$$
(22)

*Proof.* First, we use Lemma 1 followed by (21) to estimate

$$\kappa_{\mathrm{T}}(L\Sigma) \|\mathbf{x} - \mathbf{x}^*\|_L^2 \le \kappa_{\mathrm{T}}(L\Sigma, ML^{-1}) \|\mathbf{x} - \mathbf{x}^*\|_M^2$$
$$\le 2\kappa_{\mathrm{T}}(L\Sigma, ML^{-1})(f(\mathbf{x}) - f(\mathbf{x}^*))$$

Now we can just apply Lemma 6 to estimate the single step progress as

$$\mathbb{E}\left[f(\mathbf{x}_{+}) - f(\mathbf{x}^{*}) \mid \mathbf{x}\right], \leq \left(1 - \frac{h\mu}{n} + \frac{h^{2}\mu\kappa_{\mathrm{T}}(L\Sigma, ML^{-1})}{n}\right)\left(f(\mathbf{x}) - f(\mathbf{x}^{*})\right)$$

By setting  $h^{-1} = 2\kappa_{\rm T}(L\Sigma, ML^{-1})$ , the term in the left bracket becomes  $\left(1 - \frac{\mu}{4n\kappa_{\rm T}(L\Sigma, ML^{-1})}\right)$ and the proof continues as the proof of Theorem 2.

### 4.3 Discussion of the Results

The presented theoretical results extend our previous work in [32] in two ways: (i) the analysis in [32] considered only F-RP with covariance  $\Sigma = I_n$  the *n*-dimensional identity matrix with less expressive quadratic lower and upper bound assumptions; (ii) the lower- and upper bounds introduced in Section 3.2 allow for a more detailed description of the convergence rates because the quadratic model captures the eigenspectra of the functions.

We see in Theorem 2 that the number of iterations of F-RP algorithm to reach a target accuracy is proportional to  $\mu^{-1}$ . This means that for instance for  $\mu = \frac{1}{2}$ , only twice as many iterations are necessary to reach the same accuracy as with the choice  $\mu = 1$ , respectively.

The results from the Theorems 1–3 can also be extended to accommodate for more general line search oracles, for instance also with *additive error* of a constant  $\epsilon > 0$  in every step. Such errors are not crucial, the additional error terms just have to be carried along. We refer the interested reader to [32], where such analysis has been carried out for a similar problem. For functions that admit linear convergence (i.e. Theorem 2 and 3), these errors add up to an absolute constant  $C(\epsilon) = \Theta(\epsilon)$  that does not depend on the number N of iterations. On convex functions as treated in Theorem 1, the error grows as  $\epsilon N$  with the number of iterations, leading to divergence if the number N of iterations is too large. Therefore we see, that it is much better to express the errors in terms of the relative parameter  $\mu$  instead of absolute values.

All results can also be generalized to the case when the accuracy (the relative  $\mu$  and possible additive  $\epsilon$ ) of the line search oracles changes in every iteration. This amounts to different bounds on the single step progress in every iteration, and the summation in the proofs of Theorems 1–3 becomes slightly more involved (see e.g. [30]).

As a last extension, we would like to point out that the results can also be generalized to different sampling distributions and are not only valid for  $\bar{\mathcal{N}}(\mathbf{0}, \Sigma)$ vectors. However, the actual bounds on the convergence rate may change, depending on the new distribution. To determine the new convergence factors, one only has to calculate the expectation  $E_L$  in (15) for the new distribution, the rest of the proof remains the same.

### 4.4 Illustration of the results

Let us illustrate the derived bounds with an example. For simplicity, we consider a quadratic function  $f(\mathbf{x}) = \frac{1}{2} \|\mathbf{x}\|_A^2$ . Clearly,  $f \in C_A^1$  and f is strongly convex with parameter A. The algorithm F-RP with covariance  $\Sigma = I_n$  converges on this function according to Theorem 2, and the convergence rate is described by the convergence factor  $\hat{\rho}$  from (20). For exact line search ( $\mu = 1$ ) we have  $\hat{\rho} = (1 - \frac{1}{n\kappa_{\mathrm{T}}(A,A)}) \leq (1 - \frac{\lambda_{\min}(A)}{\mathrm{Tr}[A]})$ , where the last estimate follows from Lemma 4. We see that this is an improvement over the factor  $(1 - \frac{1}{n\kappa(A)})$  derived in [32] if the average of the eigenvalues of A is much smaller than the maximal one.

To demonstrate this, let us consider a class of quadratic functions,  $g_i \colon \mathbb{R}^n \to \mathbb{R}$  for  $1 \leq i < n$ , with parameter  $\ell \geq 1$ :

$$g_i(\mathbf{x}) = \frac{\ell}{2} \sum_{j=1}^{i} x_i^2 + \frac{1}{2} \sum_{j=i+1}^{n} x_i^2$$

The Hessians of all functions  $g_i$  have the same maximal ( $\ell$ ) and minimal (1) eigenvalues. The functions have two different scales that are distributed among the dimensions according to the parameter *i*. A previous numerical study [31] suggests that function  $g_i$  is challenging for RP algorithms if *i* is large (here we use  $g_{\lceil \frac{n}{2} \rceil}$  as in [31]), and easy for *i* small (here we use  $g_5$ ). Figure 2 shows the numerically observed convergence rates (black lines) of F-RP with exact line search for functions  $g_{25}$  and  $g_5$  with  $\ell = 1000$  in dimension n = 50.

The algorithm F-RP with  $\Sigma = I_n$  converges on both functions where the convergence rate can be estimated by the convergence factor (cf. Theorem 2). For both functions,  $g_{25}$  and  $g_5$ , the result established in [32] provides the same upper bound  $\left(1 - \frac{1}{\ell_n}\right)$  on the convergence factor. Our new result provided in Theorem 2 yields two different estimates for these two functions, namely  $\left(1 - \frac{2}{\ell_n}\right)$  for  $g_{25}$  and roughly  $\left(1 - \frac{1}{5\ell}\right)$  for  $g_5$  (see Table 1). This is in agreement to the empirical observations, as F-RP converges faster on  $g_5$  than on  $g_{25}$ . However, the presented bounds (red lines) slightly underestimate the rate.

It is clear that our worst-case analysis cannot give accurate convergence rates on all convex functions. We can, nonetheless, give a pointer to the part

Function	Previous result [32]	New result
<i>a</i> :	$1 - \frac{1}{n\kappa(A)}$	$1 - \frac{\lambda_{\min}(A)}{\operatorname{Tr}[A]}$
91	$1 - \frac{1}{\ell n}$	$1 - \frac{1}{i\ell + (n-i)}$

Table 1: Theoretical convergence rates of F-RP on functions  $g_i$  from previous analysis in [32], and Theorem 2.



Figure 2: Comparison of the derived convergence rates (see Table 1) with empirical measurements on  $g_{25}$  (left) and  $g_5$  (right) with  $\ell = 1000$  in dimension n = 50. Logarithm of function value vs. number of iterations (ITS). Observed F-RP convergence (solid/black), convergence rate derived in [32] (dashed), convergence rate derived in this paper (solid/red).

of the proof of Theorem 2 where we clearly use too conservative estimates. In Equation (19) we used a crude estimation of the factor  $\rho(\mathbf{x}_k)$ . This estimate is not tight in every step k (but in the worst case), as can easily seen from Equation (14) in the proof of Lemma 4. In order to find a convergence factor that best matches the observed rates we may rather analyze an *average case* scenario, and consider the expected value of  $\sigma(\mathbf{x}_k)$  over the trajectory  $\{\mathbf{x}_k\}_{k\geq 0}$  (see Lemma 7 below). However, it seems that such an analysis is the scope of this manuscript as we would not only need the expected function values  $\mathbb{E}[f(\mathbf{x}_k)]$ ), but also precise information on  $\mathbf{x}_k$  itself. Intuitively, we would expect the iterates to be almost always at the "far ends" of the ellipsoidal level sets  $\{\mathbf{x}: \mathbf{x}^T A \mathbf{x} = c\}$  of f, and therefore (14) might be only be improved by a small factor.

**Lemma 7.** Let  $\sigma \colon \mathbb{R}^n \to \mathbb{R}_{\geq 0}$  nonnegative, let a, b > 0, let  $\{\mathbf{x}_k\}_{k\geq 0}^N$  and arbitrary sequence of N points in  $\mathbb{R}^n$  and  $\bar{\sigma} = \frac{1}{N} \sum_{i=1}^N \sigma(\mathbf{x}_k)$ . Then

$$\Pi_N := \prod_{k=1}^N \left( 1 - \frac{a}{\sigma(\mathbf{x}_k) + b} \right) \le \exp\left[ -\frac{aN}{\bar{\sigma} + b} \right].$$

*Proof.* The function  $\frac{1}{x}$  for x > 0 is convex, therefore by Jensen's inequality and  $1 - x \le e^{-x}$  we find

$$-\ln \Pi_N \ge \sum_{k=1}^N \frac{a}{\sigma(\mathbf{x}_k) + b} \ge \frac{aN}{\frac{1}{N} \sum_{i=0}^{N-1} \sigma(\mathbf{x}_k) + b} = \frac{aN}{\bar{\sigma} + b}.$$

# 5 Metric Learning in Random Pursuit

In Section 4 we have derived exact bounds on the progress rate of F-RP that depend on the sampling distribution. For a quadratic function  $f(\mathbf{x}) = \frac{1}{2} \|\mathbf{x}\|_{H}^{2}$  with  $H \in \text{PD}_{n}$ , the expected running time of F-RP with search directions  $\mathbf{u} \sim \overline{\mathcal{N}}(\mathbf{0}, I_{n})$  is  $O(\kappa_{\mathrm{T}}(H)n \ln \frac{1}{\epsilon})$ , where  $\epsilon > 0$  is the desired accuracy. In contrast, for  $\mathbf{u} \sim \overline{\mathcal{N}}(\mathbf{0}, H^{-1})$  the running time drops to  $O(n\frac{1}{\epsilon})$ . This rate is (i) independent of the function f (i.e., the spectrum of H) and (ii) optimal from a theoretical point of view. This follows from the fact that  $(1 - \frac{1}{n})$  is a lower bound on the convergence rate of the "hit-and-run" scheme analyzed there is identical to the Random Pursuit.

Computing an approximation  $\hat{H}$  to H with  $\kappa_{\mathrm{T}}(\hat{H}^{-1}H) \approx 1$  and then sampling from  $\bar{\mathcal{N}}(\mathbf{0}, \hat{H}^{-1})$  instead for the optimization phase, can reduce the running time to  $O(T + n \ln \frac{1}{\epsilon})$ , where T denotes the running time of the Hessian Estimation scheme. This Variable Metric Random Pursuit algorithm (V-RP) improves over F-RP if  $T \leq \kappa_{\mathrm{T}}(H) \ln \frac{1}{\epsilon}$ . This approach also works for general strongly convex functions  $f \colon \mathbb{R}^n \to \mathbb{R}$  where the Hessian  $\nabla^2 f(\mathbf{x})$  is not necessarily constant for all  $\mathbf{x} \in \mathbb{R}^n$ . If we assume that the Hessian is only mildly changing (see for instance Lemma 5 below) then it might suffice to find an approximation of the Hessian  $\nabla^2 f(\mathbf{x}_0)$  of the initial search point  $\mathbf{x}_0 \in \mathbb{R}^n$ . Otherwise, we should use a scheme that can iteratively update its estimation of the Hessian, allowing for unforeseen changes.

We are now left with the challenge of how to efficiently estimate a Hessian matrix H in the present gradient-free setting. Iterative stochastic covariance matrix adaptation schemes are well-established in gradient-free continuous optimization [10, 16, 20] and try to estimate directly  $\Sigma = H^{-1}$ , but are notoriously difficult to study theoretically. A welcome alternative has recently been introduced by Leventhal and Lewis [17] in form of RHE. We here review and extend their scheme. For a quadratic function  $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T H\mathbf{x}$  and initial iterate  $B_0 \in \text{PD}_n$ , Leventhal and Lewis already showed that RHE generates a random sequence  $\{B_k\}_{k\geq 0}$  of Hessian approximations with

$$\mathbb{E}[\|B_k - H\|_F^2] \le \left(1 - \frac{2}{n(n+2)}\right)^k \|B_0 - H\|_F^2.$$
(23)

Therefore, if we use RHE to generate an approximation  $\hat{H}$  of the Hessian and then use F-RP with sampling distribution  $\bar{\mathcal{N}}(\mathbf{0}, \hat{H}^{-1})$ , the running time of this two-stage V-RP algorithm is  $O(n^2 \ln \|B_0 - H\|_F + n \ln \frac{1}{\epsilon})$  on a quadratic function.

Our contributions are twofold: on the theoretical side, we provide new insights into RHE. We show that (i) RHE itself can be viewed as an instance of F-RP and (ii) give exact expressions for the expectation  $\mathbb{E}[||B_k - H||_F^2]$ . Furthermore, (iii) we estimate the impact on the running time of the aforementioned two-stage V-RP algorithm if RHE converges to  $\nabla^2 f(\mathbf{x}_0)$ , but this matrix is not a very good approximation of the Hessian at the optimum  $\mathbf{x}^*$ ,  $\nabla^2 f(\mathbf{x}_0) \neq \nabla^2 f(\mathbf{x}^*)$ . On the practical side, we (iv) present three novel and theoretically sound implementations of RHE. For many practical situations, function evaluations are the most costly operations, and the goal is to keep the number of evaluations as low as possible. The third proposed scheme allowsat the expense of  $O(n^2)$  storage—to significantly boost the performance of the RHE update in this scenario.

### 5.1 Variable Metric update scheme

RHE from Leventhal and Lewis [17] comprises direct updates of a Hessian estimate. Given a symmetric matrix  $B \in PD_n$  as current Hessian estimate, the next iterate  $B_+$  is determined according to:

$$B_{+} = B + \mathbf{u}^{T} \left( H - B \right) \mathbf{u} \cdot \mathbf{u} \mathbf{u}^{T} , \qquad (24)$$

where  $\mathbf{u} \sim \bar{\mathcal{N}}(\mathbf{0}, I_n)$ . Let us now present a novel interpretation of this update which reveals that RHE is just a special instance of a F-RP algorithm. Here, the search space of the underlying optimization problem is not  $\mathbb{R}^n$  as in Section 4, but SYM<sub>n</sub>, the space of symmetric matrices. As objective, we aim at minimizing the distance to the Hessian H, measured in the Frobenius norm:

$$g(X) := \|X - H\|_F^2 .$$
(25)

This defines a quadratic function  $g: \operatorname{SYM}_n \to \mathbb{R}$ , and for a  $B \in \operatorname{SYM}_n$  and a fixed 'search direction'  $\mathbf{u}\mathbf{u}^T$ , we can easily derive an analytic expression of  $\operatorname{LS}_q(B, \mathbf{u}\mathbf{u}^T)$ , the exact line search in direction  $\mathbf{u}\mathbf{u}^T$ . By definition, we have

$$\mathsf{LS}_g(X, \mathbf{u}\mathbf{u}^T) = \operatorname*{arg\,min}_t g(B + t\mathbf{u}\mathbf{u}^T) = \operatorname*{arg\,min}_t \left\|B + t\mathbf{u}\mathbf{u}^T - H\right\|_F^2.$$

We now determine the parameter t as to minimize the right hand side, that is  $\mathbf{u}^T B \mathbf{u} - \mathbf{u}^T H \mathbf{u} + t \mathbf{u} \mathbf{u}^T \mathbf{u} \mathbf{u}^T = 0$  and conclude

$$\mathsf{LS}_q(B, \mathbf{u}\mathbf{u}^T) = \mathbf{u}^T(H - B)\mathbf{u}.$$
 (26)

We have now established, that the RHE is (i) just an instance of a specific F-RP algorithm. Moreover, (ii) the update step in (24) corresponds to a step of F-RP with an exact line search oracle  $LS_q(B, uu^T)$ .

The formula (24), or equivalently (26), requires the evaluation of  $\mathbf{u}^T H \mathbf{u}$  with unknown H. For twice differentiable functions f the second derivative of f at  $\mathbf{x}$  in direction  $\mathbf{u}$  can be well approximated by finite differences<sup>3</sup>:

$$\mathbf{u}^{T}H\mathbf{u} \approx \frac{f(\mathbf{x} + \epsilon \mathbf{u}) - 2f(\mathbf{x}) + f(\mathbf{x} - \epsilon \mathbf{u})}{\epsilon^{2}}$$
(27)

for some small  $\epsilon > 0$  as proposed in [17]. In the convex quadratic case, the above formula is exact for arbitrary  $\epsilon > 0$ . For general functions the approximation of  $\mathbf{u}^T H \mathbf{u}$  with formula (27) may not be accurate, thus leading to a failure of the update. Note that this approach only requires two additional function evaluations at  $\mathbf{x} \pm \epsilon \mathbf{u}$ . In addition, the formula implies that the estimate  $B_+$ behaves at  $\mathbf{x}$  like the unknown Hessian along direction  $\mathbf{u}$ , that is,  $\mathbf{u}^T B_+ \mathbf{u} =$  $\mathbf{u}^T H \mathbf{u}$ . This can be seen directly from (24) by noting that  $\mathbf{u}^T \mathbf{u} \mathbf{u}^T \mathbf{u} = 1$ .

The interpretation of RHE as a F-RP algorithm allows to study inexact line search oracles, which correspond to errors in the estimation (27). Qualitative

<sup>&</sup>lt;sup>3</sup>The two points  $\mathbf{x} \pm \epsilon \mathbf{u}$  are not required to be at the same distance to  $\mathbf{x}$ . For points  $\mathbf{x} - \epsilon_1 \mathbf{u}$ ,  $\mathbf{x} + \epsilon_2 \mathbf{u}$  the curvature can be estimated by quadratic interpolation with slight adaptation of formula (27).

bounds could be derived by making strong assumptions on f, for instance assuming  $f(\mathbf{x}) := f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T H(\mathbf{x} - \mathbf{x}_0) + O(\|(\mathbf{x} - \mathbf{x}_0)\|^3)$ for every  $\mathbf{x}_0 \in \mathbb{R}^n$ , i.e. with a Hessian H independed of  $\mathbf{x}_0$ . However, this kind of very restricted objective functions are not very interesting in general. We discuss more general functions in Section 5.3 below.

#### 5.2 Convergence of Random Hessian Estimation

We now derive an exact expression for the expected single step progress of RHE.

**Lemma 8.** Let  $B, H \in \text{SYM}_n$  fixed, let  $g: \text{SYM}_n \to \mathbb{R}$  as in (25), let  $B_+$  as in (24) with  $\mathbf{u} \sim \overline{\mathcal{N}}(\mathbf{0}, I_n)$ . Then

$$\mathbb{E}\left[g(B_{+}) \mid B\right] = g(B) - \frac{2g(B) + \operatorname{Tr}[B - H]^{2}}{n(n+2)} \le \left(1 - \frac{2}{n(n+2)}\right)g(B).$$

*Proof.* By standard calculus and the definition of g we have

$$g(B_+) = g(B) - (\mathbf{u}^T (B - H)\mathbf{u})^2$$

The expectation of the second term on the right hand side was calculated in Lemma 2, and the lemma follows from the trivial estimate  $\text{Tr}[B-H]^2 \ge 0$ .  $\Box$ 

The (uniform) upper bound in Lemma 8 can be quite far away from the exact value. We estimate with Cauchy-Schwarz  $\operatorname{Tr}[B-H]^2 \leq n \|B-H\|_F^2$  and

$$g(B) - \frac{2g(B) + \operatorname{Tr}[B - H]^2}{n(n+2)} \ge \left(1 - \frac{2}{n}\right)g(B).$$

Both, the upper and lower bound on the exact factor are tight in general, but they are different by a factor of approximately n. Thus one might wonder if the result (23) from [17] is too conservative in general. But this is not the case, as we answer in the next theorem.

**Theorem 4.** [Exact RHE] Let  $H \in \text{SYM}_n$  fixed, let  $\{B_k\}_{k\geq 0}$  a sequence of iterates with  $B_{k+1} = B_k + (\mathbf{u}_k^T (H - B_k) \mathbf{u}_k) \cdot \mathbf{u}_k \mathbf{u}_k^T$  with  $\mathbf{u}_k \sim \overline{\mathcal{N}}(\mathbf{0}, I_n)$ . Denote  $X_k := B_k - H$  and let parameters  $\xi_1(k) := (\lambda_1^k + \lambda_2^k), \ \xi_2(k) := (\lambda_1^k - \lambda_2^k)$  with

$$\lambda_1 = \frac{2n^2 + 2n - 5 - \omega}{2n(n+2)}, \qquad \lambda_2 = \frac{2n^2 + 2n - 5 + \omega}{2n(n+2)},$$

and  $\omega = \sqrt{4n^2 + 4n - 7}$ . Then for N > 0

$$\mathbb{E}\left[\left\|X_{N}\right\|_{F}^{2}\right] = \xi_{1}(N)\frac{\left\|X_{0}\right\|_{F}^{2}}{2} - \xi_{2}(N)\left(\frac{(2n+1)\left\|X_{0}\right\|_{F}^{2}}{2\omega} - \frac{\operatorname{Tr}[X_{0}]^{2}}{\omega}\right),$$
$$\mathbb{E}\left[\operatorname{Tr}[X_{N}]^{2}\right] = \xi_{1}(N)\frac{\operatorname{Tr}[X_{0}]^{2}}{2} - \xi_{2}(N)\left(\frac{2\left\|X_{0}\right\|_{F}^{2}}{\omega} - \frac{(2n+1)\operatorname{Tr}[X_{0}]^{2}}{2\omega}\right).$$

Before going into the proof of this theorem, let us discuss its statement. It is not hard to see, that  $\lambda_2 \leq 1 - \frac{2}{n(n+2)}$  and  $\lambda_1 = 1 - \Theta(\frac{1}{n})$ . Therefore we can approximate the factors  $\xi_1(k) \approx -\xi_2(k) \approx \lambda_2^k$  for k large enough. The upper bound (23) from Leventhal and Lewis [17] is therefore reached if  $\operatorname{Tr}[X_0] = \operatorname{Tr}[B_0 - H] = 0$ . However, if  $|\operatorname{Tr}[X_0]|$  is large,  $\operatorname{Tr}[X_0]^2 = n ||X_0||_F^2$ , say, then term in the right bracket almost vanishes and  $\mathbb{E}[||X_k||_F^2] \approx \frac{1}{2}\lambda_2^k ||X_0||_F^2$ . Thus the estimation (23) cannot significantly be improved, regardless of  $\operatorname{Tr}[X_0]$  we have  $\frac{1}{2} ||X_0||_F^2 (1 - \frac{2}{n(n+2)})^k \lesssim \mathbb{E}[||X_k||_F^2] \leq ||X_0||_F^2 (1 - \frac{2}{n(n+2)})^k$ , where the first inequality holds up to some lower order therms of n.

**Remark 1.** The above theorem derives an exact expression for  $\mathbb{E}[||B_N - H||_F^2$ , but no high-probability estimates. With Markov's inequality one can easily get an upper bound on  $\mathbb{E}[||B_N - H||_F^2$  that holds with high probability. Let  $j \leq N$  and b > 0 with  $(1 - \frac{2}{n(n+2)})^j = b$ . We have  $\mathbb{E}[||B_N - H||_F^2 \leq (1 - \frac{2}{n(n+2)})^{N-j} ||B_0 - H||_F^2$ with probability at least 1 - b.

of Thm. 4. Let the iteration k be fixed. The exact expression for the single step progress from Lemma 8 depends not only on  $||X_k||_F^2$ , but also on  $\text{Tr}[X_k]^2$ . Let us also calculate  $\mathbb{E}[\text{Tr}[X_{k+1}]^2$ . By the definition of the update (24) we immediately get  $\text{Tr}[X_{k+1}] = \text{Tr}[X_k] - \mathbf{u}_k^T X_k \mathbf{u}_k$ , and therefore

$$\mathbb{E}\left[\operatorname{Tr}[X_{k+1}]^2 \mid \{X_i\}_{i=0}^k\right] = \operatorname{Tr}[X_k]^2 - \mathbb{E}\left[2\operatorname{Tr}[X_k]\mathbf{u}_k^T X_k \mathbf{u}_k - (\mathbf{u}_k^T X_k \mathbf{u}_k)^2 \mid X_k\right] \\ = \left(1 - \frac{2n+3}{n(n+2)}\right) \operatorname{Tr}[X_k]^2 + \frac{2}{n(n+2)} \left\|X_k\right\|_F^2,$$

with Lemma 2. We obtain a linear recurrence for the conditional expectations of  $||X_k||_F^2$  and  $\operatorname{Tr}[X_k]^2$ . What we now have to do, formally, is to condition on  $\{X_i\}_{i=0}^{k-1}$  and calculate the expectations again. By the tower property of conditional expectations,  $\mathbb{E}[E[||X_{k+1}||_F^2 | \{X_i\}_{i=0}^k] | \{X_i\}_{i=0}^{k-1}] = \mathbb{E}[||X_{k+1}||_F^2 | \{X_i\}_{i=0}^{k-1}]$ . Repeating this procedure for  $\{X_i\}_{i=0}^{k-2}$  up to  $X_0$ , we finally obtain  $E[||X_{k+1}||_F^2 | X_0] = \mathbb{E}[||X_{k+1}||_F^2]$ . We observe that all intermediate expressions only depend linearly on  $||X_0||_F^2$  and  $\operatorname{Tr}[X_0]^2$ , that is we can write  $(\mathbb{E}[||X_k||_F^2, \mathbb{E}[\operatorname{Tr}[X_k]^2])^T = C(n)^k (||X_0||_F^2, \operatorname{Tr}[X_0]^2)^T$  for a 2 × 2 matrix C(n). By linear algebra, we can now decouple the linear recurrence. This is carried out in detail in Lemma 11 in the appendix.

### 5.3 RHE on general strongly convex functions

Theorem 4 shows the convergence of RHE to one fixed target matrix  $H \in \text{PD}_n$ , where  $H = \nabla^2 f(\mathbf{x}_0)$  is the Hessian of the objective function f at a point  $\mathbf{x}_0 \in \mathbb{R}^n$ . For quadratic functions the Hessian H is constant, hence RHE converges to H regardless whether the estimates (27) are evaluated at a single point  $\mathbf{x}_0$ , or at various different points  $\{\mathbf{x}_k\}_{k\geq 0}$ . For an initial estimate  $B_0 \in \text{PD}_n$ , at most  $O(n^2 \ln ||B_0 - H||_F)$  iterations of RHE are necessary to find an approximation  $\hat{H} \approx H$ , that can be used for the sampling of the search directions in F-RP. Hence the running time of V-RP on quadratic functions is  $O(n^2 \ln ||B_0 - H||_F + n \ln \frac{1}{\epsilon})$ , where  $\epsilon > 0$  is the target accuracy. In Section 5.4 we show how this bound can be improved if we only count function evaluations, instead of iterations of RHE or F-RP.

On general strongly convex functions, the Hessian is different at every point in the space. For a fixed point  $\mathbf{x}_0 \in \mathbb{R}^n$ , Theorem 4 shows convergence of RHE to  $\nabla^2 f(\mathbf{x}_0)$  if the estimates (27) are evaluated at the single point  $\mathbf{x}_0$ . However, it might not be useful to compute an approximation of the Hessian at  $\mathbf{x}_0$  if this matrix is not close to the Hessian at the optimum  $\mathbf{x}^* \in \mathbb{R}^n$ . Therefore it seems reasonable to interlace the update steps of RHE with the search steps of F-RP, i.e. invoke one update step (24) at each search point  $\{\mathbf{x}_k\}_{k\geq 0}$  that is generated by F-RP. This approach is theoretically justified: As the iterates  $\{\mathbf{x}_k\}_{k\geq 0}$  of F-RP converge (slowly) to  $\mathbf{x}^*$ , also the corresponding Hessians  $H_k := \|\nabla^2(\mathbf{x}_k)\|$  will converge to  $H := \nabla f(\mathbf{x}^*)$ , and in [17, Theorem 2.3] it is shown, that the Hessian estimates  $\{B_k\}_{k\geq 0}$  generated by this interlaced scheme will converge to H as well. However, this theorem does not imply a strong bound on the running time, as their technique only provides a bound on the convergence factor for iterations  $k \geq K$ , where K is such that  $\|H_K - H\|_F \ll 1$ . The following example shows that K can be as large as  $O(n\kappa_{\mathrm{T}}(H))$ . Consider a strongly convex function  $f: \mathbb{R}^2 \to \mathbb{R}$  with minima  $\mathbf{x}^* = \mathbf{0}$  and Hessians

$$H(\mathbf{x}) := \begin{bmatrix} 10^9 + |x_1| & 0\\ 0 & 1 \end{bmatrix}$$

For  $\mathbf{x} \in \mathbb{R}^2$  it is required  $|x_1| < 1$  to guarantee  $||H(\mathbf{x}) - H(\mathbf{x}^*)||_F < 1$ . For initial iterate  $\mathbf{x}_0 := (10^{10}, 0)^T$ , F-RP with sampling distribution  $\overline{\mathcal{N}}(\mathbf{0}, I_2)$  needs  $O(n\kappa_{\mathrm{T}}(H(\mathbf{x}^*)))$  iterations to find such a close point. On the other hand,  $\kappa(H(\mathbf{x}_0)^{-1}H(\mathbf{x}^*)) \approx 10$ . That is, it would suffice if RHE is only invoked locally at the initial iterate  $\mathbf{x}_0$ , because an approximation  $\hat{H} \approx H(\mathbf{x}_0)$  suffices to guarantee fast convergence of F-RP with sampling distribution  $\overline{\mathcal{N}}(\mathbf{0}, \hat{H}^{-1})$ . The running time of this approach is only  $O(n^2 \ln ||B_0 - H(\mathbf{x}_0)||_F + n \ln \frac{1}{\epsilon})$  instead (see Theorem 5 below).

We conclude, that the condition  $||H_K - H||_F \ll 1$  in [17, Theorem 2.3] is far too strong for what is needed here. The following theorem aims at relaxing this condition, measuring the deviation by  $\kappa(H_K^{-1}H)$  instead. That is, we here consider only the situation where the Hessian  $\nabla^2 f(\mathbf{x}_0)$  of the initial iterate  $\mathbf{x}_0$ is already close enough to  $\nabla^2 f(\mathbf{x}^*)$  and RHE is only invoked at  $\mathbf{x}_0$ , finding an approximation  $\hat{H}$  of  $\nabla^2 f(\mathbf{x}_0)$ . We give a bound on the convergence factor of F-RP with sampling distribution  $\bar{\mathcal{N}}(\mathbf{0}, \hat{H}^{-1})$ . Using the triangle inequality as in the proof of Theorem 2.3 in [17], it would also be possible to derive an analogous statement for the interlaced V-RP approach.

**Theorem 5.** Let  $0 < a \leq b$ ,  $0 \leq c < 1$ ,  $d \geq 1$  and let  $B, H, X \in PD_n$ , with  $\|Y\|_2 \leq b$  and  $\|Y^{-1}\|_2 \leq a^{-1}$  for  $Y = \{B, H, X\}$ . Here  $\|Y\|_2$  denotes the operator norm induced by the 2-norm. Let  $\|B - X\|_F \leq a^2b^{-1}c$  and  $\kappa(H^{-1}X) \leq d$ . Then  $\kappa(H^{-1}B) \leq \frac{d+c}{1-c}$ .

For  $f \in C_L^1$  and strongly convex with parameter M, we can estimate:  $\|\nabla^2 f(\mathbf{x})\|_2 \leq \lambda_{\max}(L)$  and  $\|(\nabla^2 f(\mathbf{x}))^{-1}\|_2 \leq \lambda_{\min}^{-1}(M)$ , at any  $\mathbf{x} \in \mathbb{R}^n$ . Suppose  $\mathbf{x}_0 \in \mathbb{R}^n$  is such that for  $X := \nabla^2 f(\mathbf{x}_0)$  and  $H := \nabla^2 f(\mathbf{x}^*)$ ,  $\kappa(H^{-1}B) \leq d$  for some  $d \geq 1$  and  $B \in \text{PD}_n$  an initial iterate of RHE. According to Theorem 5 it takes  $O(n^2(\ln \|B - X\|_F + \ln \frac{\lambda_{\max}(L)}{\lambda_{\min}(M)^2}))$  iterations of (RHE) to find a sufficiently close estimate  $B_K$ , s.t.  $\kappa(B_K^{-1}H) \leq 2d + 1$ , say  $(c = \frac{1}{2})$ .

of Theorem 5. We have 
$$\lambda_{\max}(X^{-1}H) = \|X^{-1}H\|_2 \le \|X^{-1}\|_2 \|H\|_2 \le a^{-1}b$$
 by

submultiplicativity and the assumptions, hence  $\lambda_{\min}(H^{-1}X) \geq ab^{-1}$ . Therefore

$$\lambda_{\min}(H^{-1}B) = \lambda_{\min}(H^{-1}X + H^{-1}(B - X))$$
  

$$\geq \lambda_{\min}(H^{-1}X) - \left\|H^{-1}(B - X)\right\|_{2}$$
  

$$\geq \frac{a}{b} - \left\|(B - X)\right\|_{F} \left\|H^{-1}\right\|_{2} \geq \frac{a}{b} - \frac{a^{2}c}{b} \left\|H^{-1}\right\|_{2},$$

with the assumed upper bound on  $||B - X||_F$ . As  $||H^{-1}||_2 \leq a^{-1}$ , we conclude  $\lambda_{\min}(H^{-1}B) \geq (1-c)ab^{-1} > 0$ . With the analogous argument  $\lambda_{\max}(H^{-1}B) \leq \lambda_{\max}(H^{-1}X) + ab^{-1}c$  and we can estimate the condition number

$$\kappa(H^{-1}B) \leq \frac{\lambda_{\max}(H^{-1}X) + \frac{ac}{b}}{\lambda_{\min}(H^{-1}X) - \frac{ac}{b}} \leq \frac{d\lambda_{\min}(H^{-1}X) + \frac{ac}{b}}{\lambda_{\min}(H^{-1}X) - \frac{ac}{b}} \,.$$

The fraction  $\frac{dx+y}{x-y}$  for x-y > 0, d, y > 0 is maximized if x is as small as possible. With the lower bound on  $\lambda_{\min}(H^{-1}X)$  we finally conclude

$$\kappa(H^{-1}B) \le \frac{ab^{-1}(d+c)}{ab^{-1}(1-c)} \,. \qquad \Box$$

# 5.4 Implementations of RHE

Now we proceed to present three implementations of RHE. One difficulty is, that the update (24) does not guarantee that the matrix  $B_+$  is positive definite. An standard result in Wedderburn [34, pg. 69] states that for  $B \in \text{PD}_n$ ,  $\mathbf{u} \in \mathbb{R}^n$ with  $\|\mathbf{u}\| = 1$ , the matrix  $B + t\mathbf{u}\mathbf{u}^T$  is positive definite if  $t^{-1} < \mathbf{u}^T B^{-1}\mathbf{u}$ . Leventhal and Lewis suggest an *ad hoc* projection of  $B_+$  onto the cone of  $\text{PD}_n$ matrices. They numerically show that this yields a practicable algorithm [17].

The projection on  $\text{PD}_n$  is only required if the current iterate  $B_+$  is needed for the sampling of the search direction, i.e.  $\mathbf{u} \sim \bar{\mathcal{N}}(\mathbf{0}, B_+^{-1})$ . The projection step is not necessary if we let the scheme run until it converges to a matrix  $\hat{H}$ that is close to the Hessian  $H \in \text{PD}_n$  (this variant is denoted as updateHess in the supporting online material [33]).

As an alternative to the projection suggested in [17], we would like propose a different one. In sub-routine updateHessCorr depicted in Figure 3 we ensure that the generated iterates are always positive definite. If T on line 3 is not positive definite (checked by Wedderburn's formula), we apply a second RHE update step in direction  $\mathbf{v}$ , where  $\mathbf{v}$  is an eigenvector of T corresponding to the smallest (hence negative) eigenvalue of  $B_+$ . By standard matrix perturbation theory, as detailed in Lemma 9 below, the twice updated matrix will be positive definite again (as H is). This scheme comes at the expense of two additional function evaluations at  $\mathbf{x} \pm \epsilon \mathbf{v}$ . The updates in line 3 and 7 can directly be implemented using the ShermanMorrison formula. This version of the VM update has already been successfully used in a recent numerical study [31].

**Lemma 9.** Let  $A \in PD_n$ ,  $\mathbf{x} \in \mathbb{R}^n$  and  $\mathbf{z}_1 \in \mathbb{R}^n$  an eigenvector corresponding to the smallest eigenvalue of  $(A - \mathbf{x}\mathbf{x}^T)$ . Then

$$B := A - \mathbf{x}\mathbf{x}^T + \left|\lambda_{\min}(A - \mathbf{x}\mathbf{x}^T)\right| \mathbf{z}_1 \mathbf{z}_1^T \in \mathrm{PD}_n.$$

$ extsf{updateHessCorr}(f, \mathbf{x}, B, \epsilon)$	$ tupdateHessStore(f, \mathbf{x}, B, \epsilon,  ext{reuse}, m)$
Output: Hessian estimate	<b>Requires</b> : Persistent storage S of size
$B_+ \in \mathrm{PD}_n$	$O(n^2)$
1 $\mathbf{u} \sim \bar{\mathcal{N}}(0, I_n)$	<b>Output</b> : Hessian estimate
2 $\Delta_{\mathbf{u}} \leftarrow \frac{f(\mathbf{x}+\epsilon\mathbf{u})-2f(\mathbf{x})+f(\mathbf{x}-\epsilon\mathbf{u})}{\epsilon^2} - \mathbf{u}^T B \mathbf{u}$	$B_+ \in \mathrm{PD}_n$
3 if $T \leftarrow B + \Delta_n \cdot \mathbf{u} \mathbf{u}^T \in \mathrm{PD}_n$ then	1 $B_+ \leftarrow \texttt{updateHess}\{\texttt{Corr}\}(f, \mathbf{x}, B, \epsilon)$
$4 \mid B_+ \leftarrow T$	2 Add $(\mathbf{u}, \mathbf{u}^T \nabla^2 f(\mathbf{x}) \mathbf{u}), (\mathbf{v}, \mathbf{v}^T \nabla^2 f(\mathbf{x}) \mathbf{v})$
else	to $S$
5 $  \mathbf{v} \leftarrow \texttt{smallestEVec}(T)$	3 if reuse then
$6 \mid \Delta_{\mathbf{v}} \leftarrow$	4 repeat <i>m</i> times
$f(\mathbf{x}+\epsilon\mathbf{v})-2f(\mathbf{x})+f(\mathbf{x}-\epsilon\mathbf{v}) = \mathbf{v}^T T \mathbf{v}$	5   foreach $(\mathbf{s}, s) \in S$ do
$= \begin{bmatrix} e^2 & v^T \\ P & (P + \Lambda & vr^T) + \Lambda & vr^T \end{bmatrix}$	6     if
$7  \Box D_+ \leftarrow (D + \Delta_{\mathbf{v}} \cdot \mathbf{v} \mathbf{v}) + \Delta_{\mathbf{u}} \cdot \mathbf{u} \mathbf{u}$	$   T \leftarrow B_+ + (s - \mathbf{s}^T B_+ \mathbf{s}) \cdot \mathbf{s} \mathbf{s}^T \in \mathrm{PD}_n $
s return $B_+$	<b>then</b> $B_+ \leftarrow T$
	7 return $B_+$

Figure 3: Two implementations of RHE (24). Left panel: The Hessian estimation B is updated in every step. Positive definiteness is established by an additional projection step. Right panel: the finite difference approximations for  $\mathbf{u}^T \nabla^2 f(\mathbf{x}) \mathbf{u}$  are stored. This information can be used for additional update steps that do not require additional function evaluations. The storage S saves the  $O(n^2)$  most recently added elements. If the capacity of S is exceeded, the oldest element is deleted (see main text for further information).

*Proof.* The matrix  $(A - \mathbf{x}\mathbf{x}^T)$  is symmetric. Let  $(A - \mathbf{x}\mathbf{x}^T) = \sum_{i=1}^n \lambda_i \mathbf{z}_i \mathbf{z}_i^T$  denote its spectral decomposition with  $\lambda_1 \leq \lambda_2 \leq \ldots \lambda_n$  in increasing order. If  $\lambda_1 \geq 0$ , then there is nothing to show. Otherwise, we observe that by a variant of Weyl's theorem (cf. [12, Theorem 4.3.4]),  $0 \leq \lambda_i(A) \leq \lambda_{i+1}(A - \mathbf{x}\mathbf{x}^T) = \lambda_{i+1}$  for  $i = 1, \ldots, n-1$ . Thus at most  $\lambda_1$  can be negative. We conclude

$$\mathbf{y}^T B \mathbf{y} = \mathbf{y}^T \left( \sum_{i=1}^n \lambda_i \mathbf{z}_i \mathbf{z}_i^T + |\lambda_1| \, \mathbf{z}_1 \mathbf{z}_1^T \right) \mathbf{y} \ge \mathbf{y}^T \left( \lambda_1 \mathbf{z}_1 \mathbf{z}_1^T + |\lambda_1| \, \mathbf{z}_1 \mathbf{z}_1^T \right) \mathbf{y} \ge 0,$$

for all  $\mathbf{y} \in \mathbb{R}^n$ .

The two implementations discussed so far need in every iteration at least two function evaluations to perform the update. In settings where function evaluations are costly or time consuming one could also store previously computed function values and reuse them for the updates. If we assume that either the RHE scheme is invoked locally at one fixed point  $\mathbf{x}_0$  (as discussed in Section 5.3), or the Hessian  $\nabla^2 f(\mathbf{x}_k)$  is only mildly changing between successive iterates  $\mathbf{x}_k$ , then the previously computed values  $\mathbf{u}_{k-t}^T \nabla^2 f(\mathbf{x}_{k-t}) \mathbf{u}_{k-t}$  back to some horizon  $h, t = 1, \ldots, h$ , might still be accurate estimates of the curvature in direction  $\mathbf{u}_{k-t}$  at the current position  $\mathbf{x}_k$ . Thus, one might apply the update (24) again for directions  $\mathbf{u}_{k-t}$  using the approximation  $\mathbf{u}_{k-t}^T H(\mathbf{x}_k) \mathbf{u}_{k-t} \approx \mathbf{u}_{k-t}^T H(\mathbf{x}_{k-t}) \mathbf{u}_{k-t}$ . This requires additional computation time but no additional function evaluations. This version of the update is presented in sub-routine updateHessCorr depicted in Figure 3 with with horizon  $h = O(n^2)$ . This variant is motivated by the following observation.

**Theorem 6.** Let  $0 < \epsilon < 1$  and constant C > 0 large enough, according to [1, Theorem 4.2] (see the proof below). Let U be a set of  $h = Cn^2$  normalized normal vectors  $\{\mathbf{u}_i\}_{i=1}^h$ ,  $\mathbf{u}_i \sim \bar{\mathcal{N}}(\mathbf{0}, I_n)$  for  $i = 1, \ldots, h$  and let  $H, B \in \text{PD}_n$ . Then for  $\mathbf{u}$  sampled uniformly at random from the (fixed) set U, denoted as  $\mathbf{u} \sim U$ , it holds for  $B_+ = B + \mathbf{u}^T (H - B) \mathbf{u} \cdot \mathbf{u} \mathbf{u}^T$ :

$$\mathbb{E}_{\mathbf{u} \sim U} \left[ \|B_{+} - H\|_{F}^{2} \right] \leq \left( 1 - \frac{(1 - \epsilon)2}{n(n+2)} \right) \|B - H\|_{F}^{2},$$

with probability at least  $1 - e^{-\sqrt{n}}$  over the choice of U.

*Proof.* As in the proof of Lemma 8 we need to give a lower bound on the expectation  $\mathbb{E}_{\mathbf{u}\sim U}[(\mathbf{u}^T X \mathbf{u})^2]$  for X = B - H. Without loss of generality we can assume  $||X||_F = 1$ . Let V denote an orthogonal matrix such that  $VXV^T$  is diagonal, with the vector of eigenvalues  $\lambda \in \mathbb{R}^n$  on its diagonal. By considering the set  $U' := \{V\mathbf{u}: \mathbf{u} \in U\}$  instead of U, we can therefore also assume that X is diagonal and write

$$\mathbb{E}_{\mathbf{u}\sim U'}\left[(\mathbf{u}^T X \mathbf{u})^2\right] = \mathbb{E}_{\mathbf{u}\sim U'}\left[\sum_{i=1}^n \lambda_i^2 u_i^4 + \sum_{i\neq j} \lambda_i \lambda_j u_i^2 u_j^2\right],\qquad(28)$$

where the subscripts  $u_i, \lambda_i$  denote the *i*-th entry of the vectors **u** or  $\boldsymbol{\lambda}$ . By Lemma 2 (just set  $A = \mathbf{e}_i \mathbf{e}_i^T$  and  $\mathbf{x} = \mathbf{e}_j$ , where  $\mathbf{e}_i$  denotes the standard unit vector) we have

$$\mathbb{E}_{\mathbf{u}\sim\bar{\mathcal{N}}(\mathbf{0},I_n)}[u_i^4] = \frac{3}{n(n+2)}, \qquad \mathbb{E}_{\mathbf{u}\sim\bar{\mathcal{N}}(\mathbf{0},I_n)}[u_i^2 u_j^2] = \frac{1}{n(n+2)},$$

for  $i \neq j$ . We will now show that the sample approximation, i.e. the expectation over the set U', approximates these values very well with high probability. Theorem 4.2 from [1] states that for  $\epsilon > 0$ , there exist a C > 0 (depending only on  $\epsilon$ ), such that with probability at least  $1 - e^{-\sqrt{n}}$ 

$$\sup_{\mathbf{y}\in S^{n-1}} \left| \mathbb{E}_{\mathbf{u}\sim U'} \left[ \langle \mathbf{u}, \mathbf{y} \rangle^4 \right] - \mathbb{E}_{\mathbf{u}\sim \bar{\mathcal{N}}(\mathbf{0}, I_n)} \left[ \langle \mathbf{u}, \mathbf{y} \rangle^4 \right] \right| \le \frac{\epsilon}{n(n+2)}, \quad (29)$$

if the sample size  $h \ge Cn^2$ . Here we have chosen h large enough s.t. (29) holds with probability at least  $1 - e^{-\sqrt{n}}$ . Hence, the choice  $\mathbf{y} = \mathbf{e}_i$  gives an estimate of the fourth moment of  $u_i$ , i.e.

$$\frac{3-\epsilon}{n(n+2)} \le \mathbb{E}_{\mathbf{u}\sim U'}[u_i^4] \le \frac{3+\epsilon}{n(n+2)}.$$
(30)

The choice  $\mathbf{y} = \frac{1}{\sqrt{2}} (\mathbf{e}_i \pm \mathbf{e}_j)$  in (29) gives us the estimates

$$\frac{12-4\epsilon}{n(n+2)} \leq \mathbb{E}_{\mathbf{u}\sim U'}[u_i^4 + u_j^4 + 6u_i^2u_j^2 \pm 4(u_iu_j^3 + u_i^3u_j)] \leq \frac{12+4\epsilon}{n(n+2)}$$

Adding these two bounds eliminates the last two terms with odd exponents and by subtracting the estimates of  $\mathbb{E}_{\mathbf{u}\sim U'}[u_i^4]$  and  $\mathbb{E}_{\mathbf{u}\sim U'}[u_i^4]$ , we get

$$\frac{1-\epsilon}{n(n+2)} \le \mathbb{E}_{\mathbf{u}\sim U'}[u_i^2 u_j^2] \le \frac{1+\epsilon}{n(n+2)} \,. \tag{31}$$

Using (30) and (31) in (28), we get the lower bound

$$\mathbb{E}_{\mathbf{u}\sim U'}\left[(\mathbf{u}^T X \mathbf{u})^2\right] \ge (1-\epsilon) \mathbb{E}_{\mathbf{u}\sim \bar{\mathcal{N}}(\mathbf{0}, I_n)}\left[(\mathbf{u}^T X \mathbf{u})^2\right] \,,$$

and the theorem follows from Lemma 8 which provides a lower bound on  $\mathbb{E}_{\mathbf{u}\sim\bar{\mathcal{N}}(\mathbf{0},I_n)}\left[(\mathbf{u}^T X \mathbf{u})^2\right]$ .

# 5.5 Computational Experiments

To complement our theoretical investigation, we conducted a few numerical experiments. We compared the here presented V-RP variants with a number of randomized and deterministic derivative-free algorithms. The set of benchmark functions comprised three quadratic, and one non-convex function.

#### 5.5.1 Benchmark setting

The tested variants of V-RP comprise both implementations of RHE that were presented in Figure 3, in combination with three different implementation of the line search oracle: (i) MATLAB's built-in routine fminunc.m, (ii) an exact line search that needs only two additional function evaluations (three in total on the line) on quadratic functions, by interpolation, and (iii) adaptive step size control from Evolutionary Computation. This last scheme only proves one additional point along the chosen line. The new point is accepted if the function value of the new point is lower than the function value of the previous iterate. In order to ensure that a positive fraction p is accepted on average we use the subroutine **aSS** detailed in Figure 1 from [31] with parameters p = 0.27and  $\sigma = 1$ . We used the following schemes for our comparison: The Evolution Strategy with Covariance Matrix Adaptation and mirrored sampling and sequential selection (CMA-ES) [10, 4]; Implicit Filtering (IMFILL) [15]; the classical Down-Hill Simplex algorithm (Nelder-Mead) [21], the accelerated version of Nesterov's [22] gradient-free Random Gradient algorithm (Nesterov acc.); Powell's NEWUOA[24]; and Pattern-Search that is available in MATLAB. The full description of the algorithms, as well as the details regarding the parameter selection, we refer to the supporting online material [33] where the benchmark setting is presented in full detail.

To evaluate the power of adaptation, we tested the algorithms on the following parametric set of functions with increasing curvature. We consider

$$f_1(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^n e^{1 + (i-1)\frac{\log \ell - 1}{n-1}} x_i^2,$$

with parameters  $\ell = 10^i$  for i = 0, ..., 7. To test the "valley-following" abilities of the different algorithms we also include the non-convex Rosenbrock [27] function  $f_2$  in the benchmark set:

$$f_2(\mathbf{x}) = \sum_{i=1}^{n-1} \left( 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right) \,.$$

In order to prohibit the tested algorithms from making use of the diagonal structure of the Hessian matrices of  $f_1$  we rotate the function domain by generating random rotation matrices R with  $RR^T = I_n$  and a shift parameter

l	Nestero	v NEW-	IMFIL	Nelder-	Pattern	CMA-	V-RP	V-RP
	Acc.	UOA				$\mathbf{ES}$	(corr)	(store)
				Mead			exact	exact
0	3.96	0.70	4.08	-	66.35	3.51	5.08	4.87
1	9.77	0.81	7.39	-	103.61	3.83	6.14	5.20
2	37.54	2.22	8.25	-	-	6.06	10.12	7.70
3	138.25	6.74	10.60	-	-	11.20	17.42	9.38
4	-	20.03	-	-	-	18.56	27.41	10.32
5	-	51.14	-	-	-	26.45	38.76	12.33
6	-	105.19	-	-	-	34.41	50.05	15.52
7	-	-	-	-	-	42.82	62.44	19.44

Table 2: Number of FES/ $n^2$  to reach accuracy  $10^{-8}$  on  $f_1$  with parameter  $\ell$ , in n = 20 dimensions (mean over 31 independent runs). A dash '-' indicates that accuracy could not be reached within a budget of  $200n^2$  FES. V-RP equipped with an exact line search oracle.

acc.	F-RP	NEW-	IMFIL	Nelder-	Pattern	CMA-	V-RP	V-RP
		UOA				$\mathbf{ES}$	(corr)	(store)
				Mead			ES	ES
$10^{1}$	57.21	4.73	-	-	139.26	10.36	24.83	22.95
$10^{0}$	186.64	10.14	-	-	-	26.30	49.79	44.52
$10^{-2}$	-	12.94	-	-	-	31.76	64.52	56.18
$10^{-4}$	-	14.81	-	-	-	33.47	70.26	61.02
$10^{-6}$	-	16.29	-	-	-	34.76	73.31	63.23
$10^{-8}$	-	17.55	-	-	-	35.96	75.56	65.16
sec.	32.90	6.55	8588.06	14.38	1979.45	13.96	11.62	140.89

Table 3: Reached accuracy vs. number of  $FES/n^2$  on  $f_2$  in n = 20 dimensions (mean over 31 independent runs). A dash '-' indicates that accuracy could not be reached within a budget of  $200n^2$  FES. V-RP equipped with adaptive step size control, see [33].

 $\mathbf{x}_s \sim \mathcal{N}(0, I_n)$ , thus leading to function instances of the form  $f(R(\mathbf{x} - \mathbf{x}_s))$ . We also apply the same transformation and shift to the initial iterate  $\mathbf{x}_0$ , which is  $\mathbf{x}_0 = \mathbf{1}_n$  for  $f_1$  and  $\mathbf{x}_0 = \mathbf{0}_n$  for  $f_2$ .

#### 5.5.2 Computational results

We report the average number of function evaluations (FES) needed to reach accuracy  $10^{-8}$  on each function (for 31 independent trials). A summary of the collected data on  $f_1$  for all parameters  $\ell$  is presented in Table 2. Table 4 in the appendix shows more details for  $\ell = 10^5$ . We observe that V-RP-ES (with updateHessStore outperforms all algorithms for  $\ell > 3$ . IMFIL reaches the target accuracy only for  $\ell \leq 3$ , but in this regime its efficiency is comparable to V-RP. NEWUOA is superior to V-RP for  $\ell \leq 3$ . For  $\ell \geq 4$  NEWUOA needs a rapidly increasing number of FES and can not reach the target accuracy for  $\ell = 7$ . CMA-ES is superior for  $\ell \leq 3$  and is outperformed by V-RP for  $\ell > 3$ . Nelder-Mead fails for all settings to reach the target accuracy (its progress can be observed in Table 4. Pattern search is only successful for  $\ell \leq 1$  and needs at least a factor of 10 times more FES than all other algorithms.

The data for Rosenbrock's function is listed in Table 3. We observe that

only V-RP algorithms, CMA-ES, and NEWUOA are able to solve Rosenbrock's function  $f_4$  in n = 20 dimensions. Surprisingly, the NEWUOA outperforms both CMA-ES and all V-RP variants, not only in number of FES but also in computation time. None of the non-adaptive algorithms shows competitive performance.

In summary, our results show that only adaptive schemes like V-RP, CMA-ES and NEWUOA, are competitive algorithms in the presence of ill-conditioning.

# 6 Discussion and Conclusion

In this contribution we have analyzed Random Pursuit algorithms that employ (i) a fixed but arbitrary metric (Fixed Metric Random Pursuit) and (ii) a variable metric learning procedure (Variable Metric Random Pursuit). We have detailed convergence proofs and convergence rates for these Random Pursuit algorithms on convex functions. We have used an improved (matrix) quadratic upper bound technique to show expected single-step progress and global convergence of Fixed Metric Random Pursuit on (strictly) convex functions. We have provided exact expressions for the expected progress of the Randomized Hessian estimation scheme (RHE). We have shown that Variable Metric Random Pursuit can achieve almost optimal convergence rate on strongly convex functions that—after a finite learning phase of length at most  $O(n^2)$ —does not depend on the underlying properties of the unknown Hessian of the function. If the Hessian  $H_0$  at the initial search point is close to the Hessian H at the optimum, i.e.  $\kappa(H_0^{-1}H) \leq c$  for a constant c, it suffices to invoke RHE only once at the beginning.

The numerical experiments show that adaptive schemes are in general (condition number exceeding  $10^3$ ) superior to non-adaptive schemes. For high target accuracy, both V-RP and CMA-ES outperformed the other tested algorithms on the quadratic functions, both in terms of number of FES and time efficiency. NEWUOA shows excellent performance on the non-convex Rosenbrock function.

A number of theoretical challenges remain. For instance, it is still an open question how to analyze Random Pursuit schemes for constrained optimization problems of the form

$$\min f(x) \quad \text{subject to} \quad x \in \mathcal{C} \,, \tag{32}$$

where  $\mathcal{C} \subset \mathbb{R}^n$  is a convex set. And it is an open problem to derive convergence guarantees for Random Pursuit schemes on non-convex functions, such as, e.g., on the class of globally convex (or  $\delta$ -convex) functions [13] or on noisy functions with certain bounds on the variance of the noise. Finally, convergence on the important class of non-smooth convex functions is another fundamental challenge for gradient-free optimization that, most likely, needs novel tools and techniques to be developed by the mathematical programming community.

#### Acknowledgements

We like to thank the anonymous reviewers whose comments and suggestions very much helped to improve the quality and content of this paper.

# References

- Adamczak, R., Litvak, A.E., Pajor, A., Tomczak-Jaegermann, N.: Quantitative estimates of the convergence of the empirical covariance matrix in log-concave ensembles. Journal of the AMS 23, 535–561 (2010). DOI MR2601042
- [2] Andrieu, C., Thoms, J.: A tutorial on adaptive MCMC. Statistics and Computing 18(4), 343–373 (2008). DOI DOI10.1007/s11222-008-9110-y
- [3] Armijo, L.: Minimization of functions having Lipschitz continuous first partial derivatives. Pacific Journal of Mathematics 16(1), 1–3 (1966). URL http://projecteuclid.org/euclid.pjm/1102995080
- [4] Brockhoff, D., Auger, A., Hansen, N., Arnold, D., Hohm, T.: Mirrored Sampling and Sequential Selection for Evolution Strategies. In: PPSN XI, *LNCS*, vol. 6238, pp. 11–21. Springer (2011)
- [5] Broyden, C.G.: The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations. IMA Journal of Applied Mathematics 6(1), 76-90 (1970). DOI 10.1093/imamat/6.1.76. URL http://imamat.oxfordjournals.org/content/6/1/76.abstract
- [6] Davidon, W.C.: Variable Metric Method for Minimization. SIAM Journal on Optimization 1(1), 1-17 (1991). DOI 10.1137/0801001. URL http: //link.aip.org/link/?SJE/1/1/1
- [7] Fletcher, R.: A new approach to variable metric algorithms. The Computer Journal 13(3), 317-322 (1970). DOI 10.1093/comjnl/13.3.317. URL http: //comjnl.oxfordjournals.org/content/13/3/317.abstract
- [8] Goldfarb, D.: A Family of Variable-Metric Methods Derived by Variational Means. Mathematics of Computation 24(109), 23-26 (1970). URL http: //www.jstor.org/stable/2004873
- [9] Goldstein, A.: On Steepest Descent. Journal of the Society for Industrial and Applied Mathematics Series A Control 3(1), 147-151 (1965). DOI 10.1137/0303013. URL http://epubs.siam.org/doi/abs/10.1137/0303013
- [10] Hansen, N., Ostermeier, A.: Completely Derandomized Self-Adaption in Evolution Strategies. Evolutionary Computation 9(2), 159–195 (2001)
- [11] Heijmans, R.: When does the expectation of a ratio equal the ratio of expectations? Statist. Papers 40, 107–115 (1999)
- [12] Horn, R.A., Johnson, C.R.: Matrix analysis, reprint 1990 edn. Cambridge University Press (1985)
- Hu, T.C., Klee, V., Larman, D.: Optimization of globally convex functions. SIAM Journal on Control and Optimization 27(5), 1026–1047 (1989). DOI 10.1137/0327055. URL http://link.aip.org/link/?SJC/27/1026/1

- [14] Jägersküpper, J.: Lower bounds for hit-and-run direct search. In: J. Hromkovic, R. Královic, M. Nunkesser, P. Widmayer (eds.) Stochastic Algorithms: Foundations and Applications, *Lecture Notes in Comput. Sci.*, vol. 4665, pp. 118–129. Springer Berlin (2007)
- [15] Kelley, C.T.: Implicit Filtering. SIAM (2011)
- [16] Kjellström, G., Taxen, L.: Stochastic Optimization in System Design. IEEE Trans. Circuits Systems 28(7) (1981)
- [17] Leventhal, D., Lewis, A.S.: Randomized Hessian estimation and directional search. Optimization 60(3), 329-345 (2011). DOI 10.1080/ 02331930903100141. URL http://www.tandfonline.com/doi/abs/10. 1080/02331930903100141
- [18] Marti, K.: Controlled random search procedures for global optimization. In: V. Arkin, A. Shiraev, R. Wets (eds.) Stochastic Optimization, *Lecture Notes in Control and Information Sciences*, vol. 81, pp. 457–474. Springer (1986)
- [19] Mathai, A.M., Provost, S.B.: Quadratic forms in random variables: theory and applications. No. 126 in Statistics: textbooks and monographs. New York, Dekker (1992)
- [20] Müller, C.L., Sbalzarini, I.F.: Gaussian adaptation revisited an entropic view on covariance matrix adaptation. In: C. Di Chio et al. (ed.) EvoApplications, no. 6024 in Lecture Notes in Comput. Sci., pp. 432–441. Springer, Berlin (2010)
- [21] Nelder, J.A., Mead, R.: A Simplex Method for Function Minimization. The Computer Journal 7(4), 308-313 (1965). DOI 10.1093/comjnl/7.4.308. URL http://comjnl.oxfordjournals.org/content/7/4/308.abstract
- [22] Nesterov, Y.: Random Gradient-Free Minimization of Convex Functions. Tech. rep., ECORE (2011)
- [23] Polyak, B.: Introduction to Optimization. Optimization Software Inc, Publications Division, New York (1987)
- [24] Powell, M.: The newuoa software for unconstrained optimization without derivatives. In: G. Pillo, M. Roma (eds.) Large-Scale Nonlinear Optimization, *Nonconvex Optimization and Its Applications*, vol. 83, pp. 255-297. Springer US (2006). DOI 10.1007/0-387-30065-1\\_16. URL http://dx.doi.org/10.1007/0-387-30065-1\_16
- [25] Puntanen, S., Styan, G.P.H., Isotalo, J.: Matrix Tricks for Linear Statistical Models: Our Personal Top Twenty. Springer Berlin Heidelberg (2011)
- [26] R2012a, M.: http://www.mathworks.ch/help/toolbox/optim/ug/fminunc.html
- [27] Rosenbrock, H.H.: An automatic method for finding the greatest or least value of a function. The Computer Journal 3(3), 175-184 (1960).
   DOI 10.1093/comjnl/3.3.175. URL http://comjnl.oxfordjournals. org/content/3/3/175.abstract

- [28] Schumer, M., Steiglitz, K.: Adaptive step size random search. Automatic Control, IEEE Transactions on 13(3), 270–276 (1968). DOI 10.1109/TAC. 1968.1098903
- [29] Shanno, D.F.: Conditioning of Quasi-Newton Methods for Function Minimization. Mathematics of Computation 24(111), 647-656 (1970). URL http://www.jstor.org/stable/2004840
- [30] Stich, S.U.: Convex optimization with random pursuit. Ph.D. thesis, ETH Zurich (2014)
- [31] Stich, S.U., Müller, C.L.: On spectral invariance of randomized hessian and covariance matrix adaptation schemes. In: C. Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia, M. Pavone (eds.) Parallel Problem Solving from Nature - PPSN XII, *Lecture Notes in Computer Science*, vol. 7491, pp. 448–457. Springer Berlin Heidelberg (2012)
- [32] Stich, S.U., Müller, C.L., Gärtner, B.: Optimization of convex functions with Random Pursuit. SIAM Journal on Optimization 23(2), 1284–1309 (2013)
- [33] Stich, S.U., Müller, C.L., Gärtner, B.: Supporting online material for: Variable metric Random Pursuit. arXiv:1210.5114 (2014)
- [34] Wedderburn, J.H.M.: Lectures on Matrices (Colloquium Publications). AMS, New York (1938)
- [35] Wolfe, P.: Convergence conditions for ascent methods. SIAM Review 11(2), 226–235 (1969). DOI 10.1137/1011036
- [36] Wolfe, P.: Convergence conditions for ascent methods. II: Some corrections. SIAM Review 13(2), 185–188 (1971). DOI 10.1137/1013035

# A Appendix

# A.1 Proof of Lemma 2

*Proof.* Let  $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, I_n)$  and let  $C \in \text{PD}_n$  with  $C^2 = \Sigma$ . The random vector  $C\mathbf{u}$  is  $\mathcal{N}(\mathbf{0}, \Sigma)$  distributed and hence  $\mathbf{w} := C\mathbf{u}/||C\mathbf{u}||_{\Sigma^{-1}} = C\mathbf{u}/||\mathbf{u}||_2$  has the same distribution as  $\mathbf{v}$  by definition of the normalized distribution. Substituting  $\mathbf{v}$  by  $\mathbf{w}$ , we obtain expressions that depend only on  $\mathbf{u}$ , more precisely, ratios  $R_i(\frac{\mathbf{u}}{||\mathbf{u}||_2})$  for  $i = 1, \ldots, 5$  with powers of  $||\mathbf{u}||_2$  in the denominator. For instance, the first and the last one read as:

$$\mathbf{v}\mathbf{v}^{T} = \frac{C\mathbf{u}\mathbf{u}^{T}C}{\|\mathbf{u}\|_{2}^{2}} =: R_{1}\left(\frac{\mathbf{u}}{\|\mathbf{u}\|_{2}}\right), \quad \|\langle \mathbf{x}, \mathbf{v} \rangle \mathbf{v}\|_{A}^{2} = \frac{\|\langle C\mathbf{x}, \mathbf{u} \rangle \mathbf{u}\|_{CAC}^{2}}{\|\mathbf{u}\|_{2}^{4}} =: R_{5}\left(\frac{\mathbf{u}}{\|\mathbf{u}\|_{2}}\right)$$

Let  $S, T: \mathbb{R}^n \to \mathbb{R}$  denote two measurable functions in the random variable **u**. We write S and R for short to denote  $S(\mathbf{u})$  and  $T(\mathbf{u})$  respectively. Lemma 1 from [11] shows that  $\mathbb{E}\begin{bmatrix}S\\T\end{bmatrix} = \frac{\mathbb{E}[S]}{\mathbb{E}[T]}$  if and only if the covariance  $\operatorname{cov}\left(\frac{S}{T},T\right) = 0$ . This follows immediately from  $\operatorname{cov}\left(\frac{S}{T},T\right) = \mathbb{E}[V] - \mathbb{E}\begin{bmatrix}S\\T\end{bmatrix}\mathbb{E}[T]$ . We will now apply this result here. The functions  $R_i$  for  $i = 1, \ldots, 5$  do only depend on the direction of the vector  $\mathbf{u}$ , but not on its norm. Hence,  $R_i$  and  $\|\mathbf{u}\|_2$  are independent, and especially uncorrelated.

This means that we can calculate the expectations of the numerators and denominators in  $R_i$  for i = 1, ..., 5 separately. These values for the numerators follow directly from Lemma 10, and for the denominators we use  $\mathbb{E}[\|\mathbf{u}\|_2^2] = n$  and  $\mathbb{E}[\|\mathbf{u}\|_2^4] = n(n+2)$ , two well-known properties of  $\chi^2$ -distributed random variables, see e.g. [19, Thm. 3.2b.2].

The following lemma summarizes some facts about moments of quadratic forms in multivariate normal random variables.

**Lemma 10.** Let  $\mathbf{u} \in \mathcal{N}(\mathbf{0}, \Sigma)$  be drawn from the multivariate normal distribution over  $\mathbb{R}^n$  with covariance  $\Sigma \in \mathrm{PD}_n$ , and let  $A \in \mathrm{SYM}_n$  be a symmetric  $n \times n$  matrix. Then

$$\mathbb{E}[\mathbf{u}\mathbf{u}^T] = \Sigma, \quad \mathbb{E}[\mathbf{u}^T A \mathbf{u}] = \operatorname{Tr}[A\Sigma], \quad \mathbb{E}[(\mathbf{u}^T A \mathbf{u})^2] = \operatorname{Tr}[A\Sigma]^2 + 2\operatorname{Tr}[(A\Sigma)^2],$$

and for  $\mathbf{x} \in \mathbb{R}^n$ ,

$$\mathbb{E}\left[\langle \mathbf{x}, \mathbf{u} \rangle \, \mathbf{u}\right] = \Sigma \mathbf{x} \,, \quad and \quad \mathbb{E}\left[\left\|\langle \mathbf{x}, \mathbf{u} \rangle \, \mathbf{u}\right\|_{A}^{2}\right] = \operatorname{Tr}[A\Sigma] \, \|\mathbf{x}\|_{\Sigma}^{2} + 2 \, \|\mathbf{x}\|_{\Sigma A\Sigma}^{2} \,.$$

The first claim immediately follows from the definition and the second and fourth are consequences of it. This can be seen by applying linearity of expectation to the two identities  $\mathbf{u}^T A \mathbf{u} = \text{Tr}[\mathbf{u}\mathbf{u}^T A]$  and  $\langle \mathbf{x}, \mathbf{u} \rangle \mathbf{u} = \mathbf{u}\mathbf{u}^T \mathbf{x}$ . To prove the third and fifth equalities directly, one has again to use linearity of expectation, but also the fourth-moments of normal random variables. We omit the this presentation here, as the claims also follow from [19, Thm. 3.2d.3] (with the choice  $A_1 = A$  and  $A_2 = \mathbf{x}\mathbf{x}^T$  for the last claim).

# A.2 Matrix diagonalization

**Lemma 11.** Let  $n \ge 1$  and consider the following  $2 \times 2$  matrix:

$$C(n) := \begin{bmatrix} 1 - 2\eta & -\eta \\ 2\eta & 1 - (2n+3)\eta \end{bmatrix}$$

where  $\eta = \frac{1}{n(n+2)}$ . Then

$$C(n) = \begin{bmatrix} \frac{2n+1-\omega}{4\omega} & \frac{2n+1+\omega}{4\omega} \\ \frac{1}{\omega} & \frac{1}{\omega} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} -2 & \frac{\omega+2n+1}{2} \\ 2 & \frac{\omega-2n-1}{2} \end{bmatrix},$$

with  $\omega = \sqrt{4n^2 + 4n - 7}$ ,

$$\lambda_1 = \frac{2n^2 + 2n - 5 - \omega}{2n(n+2)}, \qquad \lambda_2 = \frac{2n^2 + 2n - 5 + \omega}{2n(n+2)}.$$

*Proof.* The claim can be verified by calculating the product of the three matrices.  $\hfill \Box$ 

### A.3 Additional empirical data

acc.	Nestero	v NEW-	IMFIL	N-M	Pattern	CMA-	V-RP	V-RP
	Acc.	UOA				$\mathbf{ES}$	(corr)	(store)
							exact	exact
$10^{4}$	2.21	0.09	1.59	0.86	2.87	0.71	0.27	0.20
$10^{3}$	79.53	0.33	5.95	2.83	59.55	2.87	1.59	1.30
$10^{2}$	-	1.44	54.07	18.63	-	7.18	6.79	5.56
$10^{0}$	-	14.35	-	-	-	16.35	22.97	8.16
$10^{-2}$	-	29.21	-	-	-	21.87	30.26	9.20
$10^{-4}$	-	38.96	-	-	-	24.50	34.22	10.25
$10^{-6}$	-	45.76	-	-	-	25.52	36.76	11.26
$10^{-8}$	-	51.14	-	-	-	26.45	38.76	12.33
sec.	21.96	50.67	8273.24	16.28	1863.09	10.08	7.15	26.35

Table 4: Reached accuracy vs. number of  $\text{FES}/n^2$  on  $f_1$  with parameter  $\ell = 10^5$  in n = 20 dimensions (mean over 31 independent runs). A dash '-' indicates that accuracy could not be reached within a budget of  $200n^2$  FES. Average computation time of a single run on a single core CPU: either the time until the budget of  $200n^2$  FES is exceed or the time needed to reach accuracy  $10^{-8}$ .

SUPPORTING ONLINE MATERIAL FOR

# Variable Metric Random Pursuit\*

S. U. STICH<sup>†</sup>, C. L. MÜLLER<sup>‡</sup>, AND B. GÄRTNER<sup>§</sup>

# **B** Computational Experiments

In [31] we have already presented extensive numerical results of V-RP in comparison with other randomized variable metric schemes. There, we analyzed the influence of the Hessian eigenvalue )]spectrum on the convergence of these schemes. The main result from these tests were that among a parametrized set of Hessian matrices with equal trace and condition number, the matrices with a sigmoidal spectrum are the most difficult to learn for the RHE update scheme and matrices with an inverse sigmoidal (almost flat) distribution of eigenvalues are easier to learn.

We here compare the performance of V-RP with a number of randomized and *deterministic* derivative-free algorithms. The set of test functions comprises three quadratic functions (including  $f_1$ ) with different spectra and one nonconvex function ( $f_2$ ). We first present the definition of the test functions and describe the numerical performance evaluation protocol. We then detail the algorithms and their parametrization.

### **B.1** Benchmark Functions

The first two functions are  $f_1$  and  $f_2$  (see definition in Section 5.5). We consider two additional quadratic functions with parameter  $\ell \geq 1$ .

$$f_3(\mathbf{x}) = \frac{1}{2} \left( \sum_{i=1}^{\lceil \frac{n}{2} \rceil} x_i^2 + \ell \sum_{i=\lfloor \frac{n}{2} \rfloor}^n x_i^2 \right), \quad f_4(\mathbf{x}) = \frac{1}{2} \left( x_1^2 + \frac{\ell}{2} \sum_{i=2}^{n-1} x_i^2 + \ell x_n^2 \right),$$

The Hessian matrices in both functions have the same maximal  $(\ell)$  and minimal (1) eigenvalues. The function  $f_3$  has two different scales that are distributed evenly among the dimensions. The second function  $f_4$  has – for large dimension – one global scale with one small and one large eigenvalue. A previous numerical study [31] suggests that function  $f_3$  is challenging for RP algorithms and  $f_4$  is easy among all convex quadratic functions with the same condition number and trace.

The functions  $f_3$  and  $f_4$  are limit cases of these function classes and can be considered as worst and best cases.

The quadratic functions attain their minimum function value at  $\mathbf{x}^* = \mathbf{0}_n$ , the all zero vector  $(f_1(\mathbf{0}_n) = f_3(\mathbf{0}_n) = f_4(\mathbf{0}_n) = 0)$ . The Rosenbrock function is minimized at  $\mathbf{x}^* = \mathbf{1}_n$ , with  $f_2(\mathbf{1}_n) = 0$ .

In order to prohibit the tested algorithms from making use of the diagonal structure of the Hessian matrices of  $f_1$ ,  $f_3$  and  $f_4$ , we rotate the function domain by generating random rotation matrices R with  $RR^T = I_n$  and a shift parameter  $\mathbf{x}_s \sim \mathcal{N}(0, I_n)$ , thus leading to function instances of the form

$$f(R(\mathbf{x}-\mathbf{x}_s))$$
.

We also apply the same transformation and shift to the initial iterate  $\mathbf{x}_0$ . This procedure and/or the special structure of  $\mathbf{x}^*$ . We use as initial iterate  $\mathbf{x}_0 = \mathbf{1}_n$  for the quadratic functions and  $\mathbf{x}_0 = \mathbf{0}_n$  for  $f_2$ .

# **B.2** Algorithms

# B.2.1 V-RP schemes

We implemented F-RP with fixed covariance  $\Sigma = I_n$ . This scheme is simply referred to as RP. We also implemented two RHE schemes presented in this paper, updateHessCorr and updateHessStore. The parameter setting for the latter one can be found in Fig. 3 with m = 10. The choice of the parameter  $\epsilon$ does not influence the performance of the schemes on the quadratic functions. We thus set it to  $\epsilon = 1$ . For  $f_4$  we used  $\epsilon = 10^{-6}$ . For updateHessStore we apply the updates from the storage every *n*-th iteration in random order, starting after the  $n^2$ -th iteration (as soon as enough data is collected). All RP schemes have to be combined with an implementation of the line search oracle. We tested three different implementations and present them here in increasing order of function evaluations they consume.

**ES:** This scheme is also know as (1+1)-Evolution Strategy (ES). To perform the line search, only one additional point along the chosen line is probed. The scheme accepts the new point if the function value of the new point is lower than the function value of the previous iterate. In order to ensure that a positive fraction p is accepted on average we use an adaptive step size scheme (aSS) as detailed in the procedure **aSS** in Figure 1 from [31] with parameters p = 0.27 and  $\sigma = 1$ .

**Exact:** By probing two additional points on the given line (three in total), the exact minimizer can be computed if the function is quadratic  $(f_1, f_3, f_4)$ . For  $f_2$  this scheme may fail to report a better value but we observed in our experiments that the quality of the guessed minimizer is sufficient.

Matlab: We use the built-in MATLAB routine fminunc.m from the optimization toolbox [26] with optimset('TolX'=0.01) as approximate line search. In the present gradient-free setting fminunc.m uses a mixed cubic/quadratic polynomial line search where the first three points bracketing the minimum are found by bisection [26].

#### B.2.2 CMA-ES

The Evolution Strategy with Covariance Matrix Adaptation [10] (CMA-ES) is one of the most popular and efficient schemes for derivative free optimization on non-convex and noisy problems. New search points are sampled from a multivariate normal distribution whose parameter are updated in each iteration. The fundamental design principle used here is slightly different than for the V-RP schemes. Instead of performing the updates on an estimation of the Hessian (and then computing its inverse), the updates are performed directly on the inverse directly. The CMA-ES scheme is augmented by an auxiliary variable called evolution path that takes into account the correlation of successive means taken over a finite horizon. This is similar in spirit to Rao-Blackwellization techniques in Marko Chain Monte Carlo methods [2] and Polyak's heavy ball method in first-order optimization [23]. Among the many different instances of CMA-ES, we consider here the one that is the fastest scheme for quadratic functions known today. This scheme is called the (1,4)-CMA-ES with mirrored sampling and sequential selection. We also refer to [4] for a full description of this scheme and all parameter settings used. The scale parameter is set to  $\sigma = 1$  for our experiments. The code for the (1,4)-CMA-ES scheme has been retrieved from http://coco.gforge.inria.fr/doku.php?id=bbob-2010-ret

#### **B.2.3** Nesterov's Random Gradient schemes

Nesterov [22] introduced a derivative-free optimization scheme that is very similar to RP. Optimization is performed iteratively among randomly chosen lines. The optimal step size is estimated by finite-difference estimation. This scheme is called Random Gradient (RG) method. Its advantage over RP is that the finite-difference calculation needs only one additional function evaluation, and it is guaranteed to make progress in every iteration (opposed to the ES line search). One disadvantage is that the RG method needs an estimate of the curvature of the function which is not available in practice. For test purposes, we always use the correct curvature of the objective function (parameter  $\ell$ ) as input to the RG scheme.

Similar to the accelerated gradient methods for convex optimization, an accelerated version of the RG scheme is available [22]. This scheme also needs only two function evaluation per iteration and shows superior theoretical convergence properties [22].

#### B.2.4 Pattern Search

Pattern Search is a deterministic scheme that evaluates the objective function in every iteration on 2n predefined points on a stencil. We use the built-in MATLAB routine patternsearch from the Global Optimization Toolbox [26] with parameters Cache=on, InitialMeshSize=1, TolMesh=1e-20, TolX=1e-20, TolFun=1e-20.

#### B.2.5 Nelder-Mead

We use the built-in MATLAB routine fminsearch from the Optimization Toolbox [26] which implements the classical Nelder-Mead (N-M) Down-Hill Simplex algorithm [21]. We use the algorithm with parameters TolX=1e-20, TolFun=1e-20.

#### B.2.6 NEWUOA

NEWUOA [24] is an iterative algorithm that builds a quadratic model of the objective function. Steps are proposed by minimizing this model within a trust region. When the quadratic model is updated, the new model interpolates the objective function in npt points, typically npt=2n+1. We use the C implementation made available by M. Guilbert on http://www.inrialpes.fr/bipop/people/guilbert/newuoa/newuoa.html. This code is based on the original FORTRAN implementation of NEWUOA by Powell. We use the standard setting <math>npt=2n+1 and  $\rho_{beg}=1$ ,  $\rho_{end}=10^{-14}/\ell$ .

#### **B.2.7** Implicit Filtering

Implicit filtering (IMFIL) is a hybrid of a Quasi-Newton and a VM scheme. The gradients and Hessians are approximated by finite differences. We use the MAT-LAB code by Kelly [15], available on http://www4.ncsu.edu/~ctk/imfil.html with the setting smooth\_problem=1 and bscales=  $(1, 2^{-1}, \ldots, 2^{-100})$  to avoid premature convergence.

# **B.3** Convergence on the function pair $f_3/f_4$

We test the convergence of all algorithms on  $f_3$  and  $f_4$  for dimension n = 20. We performed 31 independent trials of the same experiment. We let the algorithms run until either the accuracy  $10^{-8}$  was successfully reached, or a budget of total  $200n^2$  function evaluations (FES) was consumed. In addition to the number of FES we also recorded the run time (in seconds) needed to perform a single trial. All algorithms where executed on a single core. We report the number of function evaluations performed by the algorithm to reduce the function value by one order of magnitude.

We first focus on the results for function  $f_1$  as it presents a kind of worstcase scenario for adaptive schemes. The data for the benchmark set is listed in Table 6. Table 5 shows a subset of these data, neglecting non-adaptive schemes as well as some combinations of V-RP and line search implementation. The data in Table 5 are graphically depicted in Fig. 4. We observe that among the successful algorithms (CMA-ES, V-RP ES, and V-RP Exact) CMA-ES needed about a factor of 3.4 more FES to reach accuracy  $10^{-9}$  than both V-RP schemes but only needs half the run time. The other four algorithms (NEWUOA, IMFIL, Nelder-Mead, Pattern search) only managed to reach accuracy  $10^{0} - 10^{2}$  with the same budget of FES. With the exception of Nelder-Mead their execution time is exceeding the time of CMA-ES by a factor of 14–190.

We also observe that all seven tested algorithms make rapid progress at the beginning (up to accuracy roughly  $10^2$ ). They then get either stuck or—after a learning phase—resume fast convergence toward higher accuracy levels (roughly  $10^{-2}$ - $10^{-8}$ ). These phases are typical for these kind of algorithms (see the discussion in Section C below).



Figure 4: Reached accuracy vs. number of  $\text{FES}/n^2$  on  $f_3$  with parameter  $\ell = 10^7$  in n = 20 dimensions (mean over 31 independent runs). See Tables 5 and 6. V-RP is implemented with update scheme updateHessStore for two different implementations of the line search (ES and Exact).

The empirical results on function  $f_4$  reveal several interesting features. NEWUOA,

acc.	NEWUOA	A IMFIL	N-M	Pattern	CMA-	V-RP	V-RP
					ES	ES	Exact
107	0.06	1.58	0.74	2.18	0.23	0.68	0.11
$10^{6}$	0.13	2.32	1.72	6.52	0.57	1.21	0.30
$10^{5}$	0.17	3.07	2.52	10.63	0.93	1.77	0.53
$10^{4}$	0.23	3.91	3.19	15.32	1.28	2.29	0.80
$10^{3}$	0.28	4.68	3.71	19.79	1.62	2.83	1.09
$10^{2}$	0.34	5.59	4.11	25.18	2.27	3.37	1.40
$10^{1}$	0.40	-	6.58	-	15.29	6.15	2.97
$10^{0}$	175.60	-	-	-	30.44	11.45	13.89
$10^{-1}$	-	-	-	-	45.58	13.75	17.52
$10^{-2}$	-	-	-	-	55.81	14.74	19.25
$10^{-3}$	-	-	-	-	62.74	15.68	20.07
$10^{-4}$	-	-	-	-	66.43	16.61	20.61
$10^{-5}$	-	-	-	-	69.65	17.49	21.11
$10^{-6}$	-	-	-	-	70.65	18.49	21.65
$10^{-7}$	-	-	-	-	71.81	19.42	22.17
$10^{-8}$	-	-	-	-	72.23	20.37	22.75
sec.	438.34	5711.35	10.70	1904.77	29.63	45.37	35.92

Table 5: Accuracy vs. number of  $\text{FES}/n^2$  on  $f_3$  with parameter  $\ell = 10^7$  in n = 20 dimensions (mean over 31 independent runs). A dash '-' indicates that accuracy could not be reached within a budget of  $200n^2$  FES. V-RP is implemented with update scheme updateHessStore for two different implementations of the line search (ES and Exact). Average computation time of a single run on a single core CPU; the time until the budget of  $200n^2$  FES is exceeded or the time needed to reach accuracy  $10^{-8}$ .

CMA-ES, and V-RP all show faster convergence on this function compared to  $f_3$  (in accordance with previous experiments [31]). These schemes also solve the problem to high accuracy. All other algorithms show, however, reduced performance on  $f_4$  when compared to  $f_3$  (see data in Tables 6 and 7). Both Pattern Search and Nesterov's schemes do not reach an accuracy below 10<sup>5</sup>. RP ES, IMFIL, and N-M need a considerably higher number of FES to reach an accuracy of  $10^3$ .

In summary, these results show that only adaptive schemes and, to some extent, NEWUOA, are competitive algorithms in the presence of ill-conditioning. The results also suggest that the performance of popular methods such as Implicit Filtering and Nelder-Mead (the standard derivative-free method in MAT-LAB) are not suitable *even for problems where only a few dimensions are not on the same scale* (such as  $f_4$ ).

### B.4 Evaluating the power of adaptation

Given practical limitations on the available budget of function evaluations it is natural to ask whether function evaluations should be rather spent on estimating the Hessian or for direct function optimization. In order to evaluate the power of adaptation we test the described algorithms on Rosenbrock's function and the following parametric set of functions with increasing curvature. We consider  $f_1$ with parameters  $\ell = 10^i$  for i = 0, ..., 7. For i = 0 this function equals the so-called sphere function  $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{x}$ . We report the average number of FES needed to reach accuracy  $10^{-8}$  on each function (for 31 independent trials). The

acc.	RP	NEWUO	A IMFIL	N-M	Pattern	Nesterov	Nesterov	CMA-	V-RP	V-RP	V-RP	V-RP	V-RP	V-RP
	ES					RG	Acc.	ES	(corr)	(corr)	(corr)	(store)	(store)	(store)
									ES	Exact	matlab	ES	Exact	matlab
107	0.10	0.06	1.58	0.74	2.18	0.36	0.27	0.23	0.72	0.19	0.35	0.68	0.11	0.22
10 <sup>6</sup>	0.25	0.13	2.32	1.72	6.52	0.90	0.92	0.57	1.27	0.61	1.28	1.21	0.30	0.74
$10^{5}$	0.40	0.17	3.07	2.52	10.63	1.47	2.54	0.93	1.80	0.93	2.08	1.77	0.53	1.37
$10^{4}$	0.55	0.23	3.91	3.19	15.32	2.04	3.91	1.28	2.30	1.24	2.88	2.29	0.80	1.97
$10^{3}$	0.71	0.28	4.68	3.71	19.79	2.60	5.05	1.62	2.80	1.55	3.59	2.83	1.09	2.64
$10^{2}$	0.88	0.34	5.59	4.11	25.18	3.18	-	2.27	3.37	1.84	4.51	3.37	1.40	3.49
$10^{1}$	-	0.40	-	6.58	-	3.92	-	15.29	17.93	11.47	32.47	6.15	2.97	12.42
100	-	175.60	-	-	-	-	-	30.44	34.11	44.37	86.42	11.45	13.89	30.24
$10^{-1}$	-	-	-	-	-	-	-	45.58	40.75	52.29	101.82	13.75	17.52	36.26
$10^{-2}$	-	-	-	-	-	-	-	55.81	45.26	59.92	113.60	14.74	19.25	39.83
$10^{-3}$	-	-	-	-	-	-	-	62.74	48.28	65.66	121.34	15.68	20.07	41.49
$10^{-4}$	-	-	-	-	-	-	-	66.43	50.45	69.75	127.00	16.61	20.61	42.54
$10^{-5}$	-	-	-	-	-	-	-	69.65	52.16	72.68	131.63	17.49	21.11	43.62
$10^{-6}$	-	-	-	-	-	-	-	70.65	53.72	74.87	135.02	18.49	21.65	44.86
$10^{-7}$	-	-	-	-	-	-	-	71.81	54.99	76.78	137.87	19.42	22.17	46.00
$10^{-8}$	-	-	-	-	-	-	-	72.23	56.15	78.39	140.34	20.37	22.75	47.10
sec.	30.77	438.34	5711.35	10.70	1904.77	18.26	22.68	29.63	16.83	18.65	85.60	45.37	35.92	521.82

Table 6: Reached accuracy vs. number of  $\text{FES}/n^2$  on  $f_3$  with parameter  $\ell = 10^7$  in n = 20 dimensions (mean over 31 independent runs). A dash '-' indicates that accuracy could not be reached withing a budget of  $200n^2$  FES. See main text for further information.

data of the full benchmark set are listed in Tables 8-15. Table 2 shows a subset of algorithms including Nesterov's accelerated RG scheme.

We observe that V-RP-ES (with updateHessStore and line search ES) outperforms all algorithms for  $\ell > 3$ . Nesterov's non-adaptive accelerated RG scheme is superior to V-RP only for the isotropic case  $\ell = 0$ . IMFIL reaches the target accuracy only for  $\ell \leq 3$ , but in this regime it is more efficient than V-RP. NEWUOA is superior to V-RP for  $\ell \leq 3$ . For  $\ell \geq 4$  NEWUOA needs a rapidly increasing number of FES and can not reach the target accuracy for  $\ell = 7$ . CMA-ES is superior for  $\ell \leq 3$  and is outperformed by V-RP for  $\ell > 3$ . Nelder-Mead fails for all settings to reach the target accuracy. Pattern search is only successful for  $\ell \leq 1$  and needs at least a factor of 10 times more FES than all other algorithms.

Finally, only the V-RP algorithms, CMA-ES, and NEWUOA are able to solve Rosenbrock's function  $f_2$  in n = 20 dimensions (see Table 3 for all data). Surprisingly, the NEWUOA outperforms both CMA-ES and all V-RP variants. None of the non-adaptive algorithms shows competitive performance. Pattern search and standard RP with ES line search reach an accuracy of  $10^1$  and  $10^0$ , respectively. Implicit filtering, Nelder-Mead, and Nesterov's schemes even fail to get an accuracy of  $10^1$ .

In summary, the empirical results from the presented benchmark clearly show the superiority of adaptive schemes such as V-RP and CMA-ES.

# C RHE: llustrative numerical example

We now illustrate the typical convergence behavior of Variable Metric Random Pursuit on the challenging convex quadratic function  $f_3$ , introduced in Section B.1. This function has two different scales that need to be learned. We use parameter  $\ell = 10^7$ . The ratio of largest to smallest eigenvalue of the Hessian (i.e. the condition number) is  $10^7$ , and the global minimum of

acc.	RP	NEWUOA	A IMFIL	N-M	Pattern	Nesterov	Nesterov	CMA-	V-RP	V-RP	V-RP	V-RP	V-RP	V-RP
	ES					RG	Acc.	ES	(corr)	(corr)	(corr)	(store)	(store)	(store)
									ES	Exact	matlab	ES	Exact	matlab
106	0.01	0.05	1.26	0.29	0.01	0.35	0.25	0.01	0.02	0.01	0.02	0.02	0.01	0.02
10 <sup>5</sup>	0.05	0.08	4.45	0.35	0.71	0.94	0.42	1.06	0.10	0.10	0.08	0.11	0.03	0.20
104	8.54	1.69	37.59	4.33	-	-	-	4.97	3.79	2.48	5.08	3.99	2.41	4.88
$10^{3}$	19.20	4.03	144.48	72.65	-	-	-	6.92	7.40	5.82	11.90	7.49	5.65	12.21
$10^{2}$	29.92	6.63	-	-	-	-	-	8.50	10.20	9.08	18.22	10.59	9.16	20.04
10 <sup>1</sup>	41.29	8.95	-	-	-	-	-	10.11	12.93	12.54	24.20	12.52	12.43	26.54
100	-	11.78	-	-	-	-	-	14.99	21.38	31.35	45.85	13.64	16.29	33.48
$10^{-1}$	-	51.82	-	-	-	-	-	18.59	32.83	43.32	72.59	14.59	18.43	37.34
$10^{-2}$	-	82.25	-	-	-	-	-	19.48	36.66	51.46	89.86	15.52	19.19	38.63
$10^{-3}$	-	104.93	-	-	-	-	-	20.03	38.28	54.49	97.54	16.53	19.72	39.72
$10^{-4}$	-	113.03	-	-	-	-	-	20.56	39.45	55.86	100.07	17.49	20.25	40.86
10^5	-	117.40	-	-	-	-	-	21.10	40.56	56.95	102.38	18.43	20.80	41.98
$10^{-6}$	-	120.75	-	-	-	-	-	21.61	41.53	57.76	104.36	19.35	21.30	43.15
10^7	-	123.84	-	-	-	-	-	22.13	42.52	58.49	105.94	20.34	21.82	44.24
$10^{-8}$	-	126.26	-	-	-	-	-	22.60	43.55	59.24	107.37	21.25	22.36	45.36
sec.	30.76	274.79	7976.66	17.38	1926.26	17.36	22.94	8.90	13.13	13.86	85.21	46.66	37.94	528.10

Table 7: Reached accuracy vs. number of  $\text{FES}/n^2$  on  $f_4$  with parameter  $\ell = 10^7$  in n = 20 dimensions (mean over 31 independent runs). A dash '-' indicates that accuracy could not be reached within a budget of  $200n^2$  FES. See main text for further information.

acc.	RP	NEWUOA	IMFIL	N-M	Pattern	Nesterov	Nesterov	CMA-	V-RP	V-RP	V-RP	V-RP	V-RP	V-RP
	ES					RG	Acc.	ES	(corr)	(corr)	(corr)	(store)	(store)	(store)
									ES	Exact	matlab	ES	Exact	matlab
101	0.07	0.06	0.23	0.42	3.16	0.08	0.07	0.24	0.19	0.11	0.16	0.21	0.10	0.18
$10^{0}$	0.35	0.15	0.66	5.10	10.06	0.45	0.44	0.59	1.13	0.64	1.14	1.14	0.57	1.02
$10^{-1}$	0.67	0.22	1.09	46.34	16.86	0.87	0.86	0.95	2.04	1.19	2.28	2.06	1.10	2.09
$10^{-2}$	0.99	0.30	1.51	-	23.58	1.35	1.29	1.30	2.98	1.73	3.38	3.02	1.64	3.14
$10^{-3}$	1.32	0.37	1.96	-	30.87	1.85	1.72	1.66	3.90	2.31	4.44	3.93	2.19	4.16
$10^{-4}$	1.64	0.44	2.35	-	37.27	2.35	2.17	2.03	4.88	2.90	5.61	4.81	2.71	5.26
$10^{-5}$	1.97	0.51	2.80	-	44.71	2.86	2.62	2.39	5.80	3.46	6.80	5.78	3.24	6.42
$10^{-6}$	2.32	0.58	3.23	-	52.74	3.37	3.05	2.76	6.75	3.99	7.94	6.69	3.79	7.64
$10^{-7}$	2.66	0.64	3.66	-	59.56	3.88	3.53	3.13	7.67	4.54	9.08	7.65	4.30	8.73
$10^{-8}$	3.02	0.70	4.08	-	66.35	4.40	3.96	3.51	8.63	5.08	10.28	8.57	4.87	9.89
sec.	1.15	0.26	28.37	16.94	1292.33	17.88	NaN	1.50	2.25	1.36	44.50	21.90	2.23	135.19

Table 8: Reached accuracy vs. number of  $\text{FES}/n^2$  on  $f_1$  with parameter  $\ell = 10^0$  in n = 20 dimensions (mean over 31 independent runs). A dash '-' indicates that accuracy could not be reached within a budget of  $200n^2$  FES. See main text for further information.

acc.	RP	NEWUOA	A IMFIL	N-M	Pattern	Nesterov	Nesterov	CMA-	V-RP	V-RP	V-RP	V-RP	V-RP	V-RP
	ES					RG	Acc.	ES	(corr)	(corr)	(corr)	(store)	(store)	(store)
									ES	Exact	matlab	ES	Exact	matlab
101	0.23	0.08	0.42	1.42	12.38	0.73	0.49	0.43	0.73	0.43	0.72	0.75	0.30	0.49
100	0.54	0.20	0.98	22.41	20.49	2.00	1.20	0.79	1.79	1.14	2.10	1.86	0.90	1.49
$10^{-1}$	0.91	0.27	1.57	-	30.73	3.53	2.26	1.17	2.78	1.91	3.51	2.89	1.51	2.66
$10^{-2}$	1.27	0.36	2.17	-	41.19	5.26	3.42	1.54	3.79	2.60	4.88	3.83	2.08	3.77
$10^{-3}$	1.61	0.43	2.97	-	51.40	7.01	4.38	1.92	4.70	3.21	6.19	4.79	2.60	4.84
$10^{-4}$	1.97	0.51	3.93	-	61.55	8.84	5.39	2.30	5.68	3.83	7.47	5.71	3.12	5.98
$10^{-5}$	2.31	0.59	4.93	-	72.39	10.65	6.53	2.69	6.63	4.45	8.63	6.62	3.65	7.12
$10^{-6}$	2.63	0.67	5.91	-	82.77	12.55	7.58	3.07	7.59	5.01	9.78	7.56	4.18	8.19
$10^{-7}$	2.98	0.74	6.75	-	93.21	14.48	8.64	3.45	8.60	5.59	10.99	8.47	4.69	9.39
$10^{-8}$	3.35	0.81	7.39	-	103.61	16.42	9.77	3.83	9.53	6.14	12.15	9.35	5.20	10.48
sec.	1.09	0.27	42.87	16.93	1652.45	18.72	22.82	1.64	2.26	1.47	32.35	23.23	2.60	131.40

Table 9: Reached accuracy vs. number of  $\text{FES}/n^2$  on  $f_1$  with parameter  $\ell = 10^1$  in n = 20 dimensions (mean over 31 independent runs). A dash '-' indicates that accuracy could not be reached within a budget of  $200n^2$  FES. See main text for further information.

acc.	RP	NEWUOA	A IMFIL	N-M	Pattern	Nesterov	Nesterov	CMA-	V-RP	V-RP	V-RP	V-RP	V-RP	V-RP
	ES					RG	Acc.	ES	(corr)	(corr)	(corr)	(store)	(store)	(store)
									ES	Exact	matlab	ES	Exact	matlab
$10^{2}$	0.10	0.06	0.53	0.61	4.41	0.55	0.35	0.35	0.35	0.13	0.28	0.33	0.12	0.19
$10^{1}$	0.54	0.21	1.25	5.11	25.12	4.13	2.46	0.90	1.82	1.18	2.24	1.83	0.83	1.42
100	1.42	0.44	2.14	65.80	50.68	14.13	5.55	1.51	3.40	2.53	4.77	3.31	1.85	3.53
$10^{-1}$	2.50	0.70	3.20	-	78.55	28.72	9.23	2.15	4.75	3.98	7.39	4.21	2.93	5.55
$10^{-2}$	3.68	0.94	4.47	-	108.86	45.79	12.95	2.83	5.88	5.17	9.69	5.12	3.91	7.26
$10^{-3}$	4.91	1.17	5.91	-	-	64.04	16.96	3.41	6.98	6.26	11.70	5.95	4.68	8.67
$10^{-4}$	6.17	1.40	7.03	-	-	83.27	20.92	3.99	8.07	7.22	13.33	6.94	5.37	10.00
$10^{-5}$	7.42	1.62	7.55	-	-	102.76	24.94	4.55	8.99	8.05	14.77	7.89	5.99	11.19
$10^{-6}$	8.72	1.83	7.83	-	-	122.68	29.09	5.05	9.96	8.80	16.18	8.78	6.56	12.37
$10^{-7}$	10.10	2.04	8.07	-	-	142.71	33.35	5.56	10.88	9.48	17.55	9.68	7.14	13.54
$10^{-8}$	11.50	2.22	8.25	-	-	162.88	37.54	6.06	11.80	10.12	18.85	10.61	7.70	14.62
sec.	3.37	0.50	45.49	16.30	1863.95	17.80	23.71	2.30	2.67	2.02	43.35	28.89	10.28	383.55

Table 10: Reached accuracy vs. number of  $\text{FES}/n^2$  on  $f_1$  with parameter  $\ell = 10^2$  in n = 20 dimensions (mean over 31 independent runs). A dash '-' indicates that accuracy could not be reached within a budget of  $200n^2$  FES. See main text for further information.

acc.	RP	NEWUOA	IMFIL	N-M	Pattern	Nesterov	Nesterov	CMA-	V-RP	V-RP	V-RP	V-RP	V-RP	V-RP
	$\mathbf{ES}$					RG	Acc.	ES	(corr)	(corr)	(corr)	(store)	(store)	(store)
									ES	Exact	matlab	ES	Exact	matlab
$10^{3}$	0.03	0.06	0.56	0.33	0.56	0.27	0.22	0.14	0.12	0.05	0.08	0.10	0.05	0.07
$10^{2}$	0.38	0.17	2.30	1.95	27.54	3.63	3.17	0.97	1.27	0.81	1.52	1.17	0.58	0.97
$10^{1}$	2.12	0.61	4.64	28.07	-	27.79	13.60	2.27	3.81	3.15	4.99	3.20	2.26	4.06
100	6.65	1.38	6.51	-	-	115.70	24.95	3.76	6.45	6.36	10.71	4.15	4.80	8.52
$10^{-1}$	13.08	2.18	8.59	-	-	-	39.27	5.06	8.38	8.97	15.30	5.05	5.77	10.64
$10^{-2}$	20.44	2.90	9.11	-	-	-	51.70	6.25	9.92	10.84	18.93	6.02	6.32	11.60
$10^{-3}$	28.33	3.73	9.42	-	-	-	65.28	7.29	11.22	12.38	21.99	6.94	6.86	12.61
$10^{-4}$	36.81	4.42	9.66	-	-	-	79.90	8.30	12.31	13.62	24.44	7.87	7.36	13.78
$10^{-5}$	45.38	5.14	9.85	-	-	-	92.56	9.13	13.33	14.74	26.33	8.78	7.88	14.89
$10^{-6}$	54.13	5.70	10.13	-	-	-	107.68	9.89	14.24	15.71	28.09	9.74	8.36	15.99
$10^{-7}$	63.07	6.24	10.41	-	-	-	121.35	10.59	15.23	16.61	29.74	10.67	8.86	17.06
$10^{-8}$	72.09	6.74	10.60	-	-	-	138.25	11.20	16.15	17.42	31.22	11.60	9.38	18.21
sec.	23.01	1.61	65.92	18.51	1843.56	19.05	21.96	4.28	4.02	3.09	48.35	31.62	19.01	596.88

Table 11: Reached accuracy vs. number of  $FES/n^2$  on  $f_1$  with parameter  $\ell = 10^3$  in n = 20 dimensions (mean over 31 independent runs). A dash '-' indicates that accuracy could not be reached within a budget of  $200n^2$  FES. See main text for further information.

acc.	RP	NEWUOA	A IMFIL	N-M	Pattern	Nesterov	Nesterov	CMA-	V-RP	V-RP	V-RP	V-RP	V-RP	V-RP
	ES					RG	Acc.	ES	(corr)	(corr)	(corr)	(store)	(store)	(store)
									ES	Exact	matlab	ES	Exact	matlab
$10^{4}$	0.01	0.03	0.64	0.23	0.01	0.14	0.11	0.02	0.04	0.02	0.03	0.04	0.02	0.04
$10^{3}$	0.23	0.12	1.81	1.13	7.74	2.64	2.84	0.93	0.84	0.49	0.67	0.81	0.30	0.61
$10^{2}$	2.00	0.50	7.75	5.61	-	26.29	31.08	2.97	2.95	2.32	3.83	3.06	1.66	3.36
$10^{1}$	11.98	1.80	58.45	71.09	-	-	69.18	6.13	7.71	7.08	12.08	4.28	5.48	9.98
100	38.13	4.36	-	-	-	-	111.78	8.69	11.54	12.30	23.04	5.15	6.22	11.36
$10^{-1}$	79.47	7.32	-	-	-	-	160.50	10.91	14.12	16.49	30.71	6.06	6.70	12.36
$10^{-2}$	127.32	9.78	-	-	-	-	-	12.92	15.83	19.27	36.45	6.92	7.21	13.37
$10^{-3}$	-	12.06	-	-	-	-	-	14.31	17.16	21.34	39.90	7.83	7.74	14.39
$10^{-4}$	-	14.10	-	-	-	-	-	15.63	18.35	23.05	42.88	8.72	8.23	15.54
$10^{-5}$	-	15.80	-	-	-	-	-	16.66	19.40	24.38	45.28	9.65	8.76	16.72
$10^{-6}$	-	17.25	-	-	-	-	-	17.42	20.40	25.50	47.26	10.54	9.30	17.83
$10^{-7}$	-	18.60	-	-	-	-	-	18.01	21.36	26.51	48.93	11.49	9.81	18.92
$10^{-8}$	-	20.03	-	-	-	-	-	18.56	22.31	27.41	50.54	12.39	10.32	20.06
sec.	35.37	5.24	8588.05	17.96	1853.12	17.40	21.82	7.07	5.21	5.08	58.95	29.18	23.94	624.25

Table 12: Reached accuracy vs. number of  $FES/n^2$  on  $f_1$  with parameter  $\ell = 10^4$  in n = 20 dimensions (mean over 31 independent runs). A dash '-' indicates that accuracy could not be reached within a budget of  $200n^2$  FES. See main text for further information.

acc.	RP	NEWUOA	IMFIL	N-M	Pattern	Nesterov	Nesterov	CMA-	V-RP	V-RP	V-RP	V-RP	V-RP	V-RP
	ES					RG	Acc.	ES	(corr)	(corr)	(corr)	(store)	(store)	(store)
									ES	Exact	matlab	ES	Exact	matlab
10 <sup>5</sup>	0.01	0.01	0.76	0.14	0.01	0.06	0.06	0.01	0.02	0.02	0.02	0.02	0.01	0.02
$10^{4}$	0.16	0.09	1.59	0.86	2.87	2.01	2.21	0.71	0.62	0.27	0.44	0.60	0.20	0.33
$10^{3}$	1.19	0.33	5.95	2.83	59.55	20.91	79.53	2.87	2.38	1.59	2.79	2.19	1.30	2.58
$10^{2}$	11.48	1.44	54.07	18.63	-	-	-	7.18	7.16	6.79	11.99	4.77	5.56	11.23
$10^{1}$	88.03	5.25	-	-	-	-	-	12.53	14.00	15.21	29.17	5.74	7.58	14.94
100	-	14.35	-	-	-	-	-	16.35	18.34	22.97	44.82	6.66	8.16	16.07
$10^{-1}$	-	22.55	-	-	-	-	-	19.91	21.17	27.41	53.49	7.57	8.68	17.21
$10^{-2}$	-	29.21	-	-	-	-	-	21.87	22.87	30.26	58.76	8.52	9.20	18.31
$10^{-3}$	-	34.72	-	-	-	-	-	23.54	24.24	32.46	62.10	9.45	9.75	19.39
$10^{-4}$	-	38.96	-	-	-	-	-	24.50	25.42	34.22	64.89	10.34	10.25	20.53
$10^{-5}$	-	42.51	-	-	-	-	-	25.06	26.47	35.58	67.16	11.22	10.75	21.68
$10^{-6}$	-	45.76	-	-	-	-	-	25.52	27.47	36.76	69.11	12.11	11.26	22.88
$10^{-7}$	-	48.49	-	-	-	-	-	26.03	28.42	37.84	70.90	13.05	11.76	24.03
$10^{-8}$	-	51.14	-	-	-	-	-	26.45	29.37	38.76	72.66	14.01	12.33	25.13
sec.	34.46	50.67	8273.24	16.28	1863.09	19.17	21.96	10.08	6.69	7.15	66.13	36.11	26.35	569.45

Table 13: Reached accuracy vs. number of  $FES/n^2$  on  $f_1$  with parameter  $\ell = 10^5$  in n = 20 dimensions (mean over 31 independent runs). A dash '-' indicates that accuracy could not be reached within a budget of  $200n^2$  FES. See main text for further information.

acc.	RP	NEWUOA	M IMFIL	N-M	Pattern	Nesterov	Nesterov	CMA-	V-RP	V-RP	V-RP	V-RP	V-RP	V-RP
	ES					RG	Acc.	ES	(corr)	(corr)	(corr)	(store)	(store)	(store)
									ES	Exact	matlab	ES	Exact	matlab
106	0.01	0.01	0.87	0.04	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.01	0.02
$10^{5}$	0.10	0.08	1.57	0.65	2.37	1.76	1.57	0.45	0.60	0.19	0.26	0.56	0.11	0.23
104	0.77	0.27	4.51	1.89	31.90	17.38	-	2.20	1.93	1.07	2.26	1.98	0.80	1.79
$10^{3}$	8.45	1.15	42.37	5.98	-	170.48	-	6.29	6.40	4.41	10.98	5.08	4.26	10.04
$10^{2}$	79.40	5.11	-	35.85	-	-	-	13.25	13.16	12.54	33.67	7.20	9.16	20.72
10 <sup>1</sup>	-	19.53	-	-	-	-	-	21.68	20.56	24.92	58.73	8.21	10.67	24.05
100	-	37.01	-	-	-	-	-	26.47	25.09	33.63	74.31	9.09	11.38	25.24
$10^{-1}$	-	52.61	-	-	-	-	-	29.18	27.85	38.78	83.25	9.99	11.92	26.35
$10^{-2}$	-	65.85	-	-	-	-	-	31.09	29.72	41.81	88.14	10.87	12.41	27.41
10^3	-	74.92	-	-	-	-	-	31.90	31.25	43.96	91.83	11.80	12.95	28.41
10-4	-	82.32	-	-	-	-	-	32.64	32.40	45.62	94.67	12.70	13.45	29.53
$10^{-5}$	-	89.27	-	-	-	-	-	33.08	33.45	46.93	97.09	13.67	13.97	30.68
10^6	-	95.14	-	-	-	-	-	33.48	34.41	48.12	99.09	14.60	14.49	31.81
10^7	-	100.65	-	-	-	-	-	34.01	35.32	49.17	100.86	15.57	14.99	32.99
$10^{-8}$	-	105.19	-	-	-	-	-	34.41	36.34	50.05	102.39	16.51	15.52	34.21
sec.	30.90	194.63	7987.02	16.29	1888.71	17.42	22.54	12.99	8.47	9.21	76.31	40.34	29.91	547.15

Table 14: Reached accuracy vs. number of  $\text{FES}/n^2$  on  $f_1$  with parameter  $\ell = 10^6$  in n = 20 dimensions (mean over 31 independent runs). A dash '-' indicates that accuracy could not be reached within a budget of  $200n^2$  FES. See main text for further information.

acc.	RP	NEWUOA	A IMFIL	N-M	Pattern	Nesterov	Nesterov	CMA-	V-RP	V-RP	V-RP	V-RP	V-RP	V-RP
	ES					RG	Acc.	ES	(corr)	(corr)	(corr)	(store)	(store)	(store)
									ES	Exact	matlab	ES	Exact	matlab
106	0.08	0.07	1.58	0.56	0.95	1.44	0.96	0.30	0.48	0.11	0.13	0.52	0.12	0.15
$10^{5}$	0.54	0.23	3.85	1.52	17.25	14.66	-	1.84	1.48	0.81	1.53	1.53	0.59	1.36
104	6.50	0.94	21.60	4.43	-	142.72	-	5.64	4.94	3.52	6.83	4.53	3.14	7.41
$10^{3}$	66.49	4.03	-	17.45	-	-	-	11.49	12.13	12.30	28.55	8.04	9.57	22.97
$10^{2}$	-	17.57	-	67.45	-	-	-	20.94	19.79	24.87	59.06	9.84	12.91	32.27
101	-	42.19	-	-	-	-	-	30.81	26.41	37.43	91.47	10.91	14.50	36.40
100	-	82.48	-	-	-	-	-	36.20	31.75	46.10	110.00	11.89	15.19	37.98
$10^{-1}$	-	113.01	-	-	-	-	-	38.59	35.23	50.36	120.59	12.81	15.72	39.09
$10^{-2}$	-	133.34	-	-	-	-	-	40.11	37.23	53.69	126.24	13.78	16.25	40.10
10^-3	-	149.80	-	-	-	-	-	40.80	38.63	56.18	129.87	14.74	16.73	41.09
10^4	-	165.02	-	-	-	-	-	41.22	39.90	57.91	132.38	15.65	17.27	42.16
10^5	-	177.24	-	-	-	-	-	41.62	40.92	59.22	134.52	16.56	17.81	43.24
$10^{-6}$	-	-	-	-	-	-	-	42.01	41.92	60.42	136.33	17.49	18.31	44.39
10^7	-	-	-	-	-	-	-	42.41	42.92	61.56	137.85	18.43	18.88	45.66
$10^{-8}$	-	-	-	-	-	-	-	42.82	43.94	62.44	139.45	19.40	19.44	46.85
sec.	30.98	447.57	7705.72	17.72	1861.84	18.08	22.77	16.30	10.45	11.96	74.15	45.21	34.37	519.52

Table 15: Reached accuracy vs. number of  $\text{FES}/n^2$  on  $f_1$  with parameter  $\ell = 10^7$  in n = 20 dimensions (mean over 31 independent runs). A dash '-' indicates that accuracy could not be reached within a budget of  $200n^2$  FES. See main text for further information.

 $f_3$  is at  $\mathbf{x}^* = \mathbf{0}_n$  (where  $\mathbf{0}_n$  is the all-zeros vector) with  $f_3(\mathbf{x}^*) = 0$ . We conduct 51 runs of V-RP in n = 20 dimensions. The initial conditions are  $\mathbf{x}_0 = (1, \ldots, 1, 1/\sqrt{\ell}, \ldots, 1/\sqrt{\ell})^T$ ,  $B_0 = \frac{1}{2}\ell \cdot I_n$ . The two VM update schemes updateHessCorr and updateHessStore (see Fig. 3) are tested with the setting  $\epsilon = 1$  for both schemes. The updateHessStore scheme reuses samples from the storage S in every n-th iteration. We here report the evolution of the mean, maximum, and minimum function value vs. number of iterations (#ITS). We also calculate and report the evolution of the derived convergence factor  $\hat{\varrho}$  from Thm. 2.



Figure 5: Convergence of V-RP on  $f_3$  for updateHessCorr (blue/solid) and updateHessStore (red/dashed). Parameter  $\ell = 10^7$  in n = 20 dimensions. Upper panel: Mean and max/min (grey) function values vs. #ITS over 51 runs. Lower Panel: Mean and max/min (grey) convergence factor  $\hat{\varrho}$  vs. #ITS over 51 runs. Respective upper bounds (black/dash-dotted). See main text for further information.

On quadratic functions, a typical V-RP optimization trajectory (see [17, 31] for several examples) shows three distinct phases of convergence in function val-

ues: (i) a first short phase of rapid improvement, (ii) a metric learning phase with only marginal progress in function decrease, and (iii) a final rapid decrease in function value. In the present experiments we chose the initial iterate  $\mathbf{x}_0$ such as to minimize the first phase. This allows a clearer quantification of the length of the adaptation phase. We see that the adaptation phase lasts for roughly  $5n^2$  iterations in case of updateHessStore and  $15-18n^2$  iterations for updateHessCorr. We also visualize the derived upper bounds on the convergence factor (see Remark 1) in the lower panel of Fig. 5. For updateHessCorr the curve is plotted using b = 1, and for updateHessStore using  $\tilde{c} = 1.5$ . We see that the shape of both curves resembles the observed data. However, in both cases the theoretical bounds overestimate the empirically observed curves ("shifted" to the right). In the upper panel of Fig. 5 we depict the theoretical derived upper bound on the function value (Thm. 2). For updateHessStore the shape of this curve well matches the observed convergence. For updateHessCorr we see that the empirically observed phase transition between phase (ii) and (iii) occurs more smoothly than predicted by the theoretical bound.



Figure 6: Evolution of the spectrum of  $\Sigma = B_k^{-1}$  for VM scheme updateHessCorr on  $f_3$ . Eigenvalues vs. # ITS for 1 run. Parameter  $\ell = 10^7$  in n = 20 dimensions. Left two panels with initial setting  $B_0 = \frac{\ell}{2}I_n$ , right panel with  $B_0 = I_n$ . See main text for further information.

We also illustrate the evolution of the spectrum of the estimated inverse Hessian  $\Sigma = B_k^{-1}$  in Fig. 6 for one run with update scheme updateHessCorr. At the beginning all eigenvalues are close to  $\frac{2}{\ell}$ , as  $B_0^{-1} = \frac{2}{\ell}I_n$  (left panel). Then, about half of the eigenvalues start to increase up to 1, the other half decreases to  $\frac{1}{\ell}$ . We see that the large eigenvalues of  $\Sigma$  (or correspondingly the small eigenvalues of H) are more difficult to approximate. This takes up to 16- $18n^2$  iterations. In the right panel we depicted another run with initial matrix  $\Sigma = B_0^{-1} = I_n$ . At the beginning all eigenvalues are equal to 1. Due to the nature of the VM update scheme (rank one updates), at most one eigenvalue can become different from 1 in every iteration. Thus it takes exactly n = 20iterations until all eigenvalues are between  $10^{-7}$  and  $10^{-5}$ . From this moment, the situation is similar to the experiment in the left two panels with  $B_0 = \frac{\ell}{2}I_n$ .