Diss. ETH No. 22111

Convex Optimization with Random Pursuit

A thesis submitted to attain the degree of
Doctor of Sciences of ETH Zurich
(Dr. sc. ETH Zurich)

presented by

Sebastian Urban Stich

MSc ETH Mathematics
born on 21 May 1985
citizen of Kleinlützel SO, Switzerland

accepted on the recommendation of

Prof. Dr. Bernd Gärtner, examiner
Prof. Dr. Yurii Nesterov, co-examiner
Prof. Dr. Emo Welzl, co-examiner
Dr. Christian L. Müller, co-examiner

2014

Abstract

Optimization problems are ubiquitous in science and engineering. In this thesis, we study unconstrained black-box optimization problems that can only be accessed by an oracle that returns the function value at a query point. The theory of convex optimization problems is well-developed and such problems are typically solved with gradient-based methods. For non-convex problems, there is no unifying theoretical treatment and one has to rely on, typically gradient-free, search heuristics. Here, we analyze gradient-free optimization algorithms on convex functions.

In the first part of this thesis, we study Random Pursuit algorithms. These are iterative search schemes, where each iteration consists of two steps: (i) the generation of a (random) search direction and (ii) performing a step along this direction. We present a general framework to study such algorithms and prove convergence on smooth convex and strongly convex functions. The convergence rates depend on a sufficient decrease condition that measures the quality of the generated steps. This condition is for instance met by schemes that use a line search to generate the steps. For line search algorithms, we extend the convergence analysis to functions that are not necessarily everywhere strongly convex, but only at the optimum. Line search algorithms do not need any problem specific parameterization as input and are invariant under strictly monotone transformations of the objective functions. They thus enjoy identical convergence behavior on a wider function class. We discuss several kinds of random search directions and provide estimates for the expected convergence rates.

In the second part, we present three, at first sight seemingly unrelated, optimization algorithms that can be analyzed in the Random Pursuit framework. The examples comprise (i) solving linear systems with Kaczmarz' method and (ii) Hessian learning with Leventhal and

Lewis' estimation algorithm. Both these algorithms are instances of Random Pursuit algorithms with exact line search. We show this by demonstrating that these algorithms do only require the computation of scalar products, which in turn (iii) amounts to special Random Pursuit algorithms in Hilbert spaces, that have a simple geometric interpretation. We provide exact rates for the expected convergence. The Hessian learning scheme has a specific application: it can be used to estimate the underlying metric of an optimization problem which helps to accelerate the subsequent optimization with Random Pursuit. We do not only derive precise convergence rates, we also show that a specific implementation of this combined scheme converges equally fast on all quadratic functions, i.e. it is affine invariant.

In the last chapter, we review Nesterov's gradient-based accelerated random search scheme. Each iteration of this scheme comprises two steps: (i) a simple search step like the one in the Random Pursuit algorithms and (ii) a model building step that allows for acceleration. We show that the step (i) can harmlessly be replaced with a line search, whereas the situation in step (ii) is more delicate. We cannot show that implementing step (ii) with a line search still yields acceleration, however, the resulting scheme does not diverge and converges on quadratic functions at least as fast as the simple Random Pursuit—with the possibility to accelerate.

Zusammenfassung

Optimierungsprobleme sind allgegenwärtig in der Wissenschaft und der Technik. Wir betrachten Black-Box (schwarzer Kasten) Optimierung ohne Nebenbedingungen. Bei solchen Problemen ist die Zielfunktion *a priori* unbekannt, ihre Werte können aber für jedes fest gewählte Argument mit Hilfe eines Orakels in Erfahrung gebracht werden. Während zum Lösen konvexer Optimierungsprobleme normalerweise gradientenbasierte Suchverfahren zur Anwendung kommen, ist dieser Zugang nicht sehr erfolgreich bei nichtkonvexen Problemen. Die Theorie solcher Probleme ist weniger gut verstanden und man muss auf, meist gradientenfreie, heuristische Verfahren zurückreifen. In dieser Arbeit analysieren wir gradientenfreie Suchverfahren auf konvexen Funktionen.

Im ersten Teil dieser Arbeit untersuchen wir eine bestimmte Klasse von Verfahren, die wir "Zufallsjagd" (Random Pursuit) nennen. Dies sind iterative Suchverfahren deren Iterationen aus zwei Schritten bestehen: (i) dem Auslosen einer (zufälligen) Suchrichtung und (ii) dem Auswählen eines neuen Suchpunktes entlang dieser Richtung. Wir präsentieren eine generelle Methode um solche Verfahren zu analysieren und beweisen Konvergenz auf glatten konvexen und streng konvexen Funktionen. Die Konvergenzrate hängt von der Effektivität der einzelnen Schritte ab, die wir mit Hilfe einer ausreichenden Abstiegsbedingung (sufficient decrease) messen. Diese Bedingung wird zum Beispiel von Suchpunkten erfüllt die mittels einer Liniensuche generiert werden. Für Suchverfahren mit Liniensuche können wir die Konvergenzresultate auf Funktionen ausweiten die nicht überall, sondern nur am Optimum, streng konvex sind. Verfahren mit Liniensuche haben den Vorteil, dass sie parameterfrei arbeiten und invariant sind gegenüber monotonen Transformationen der Zielfunktion. D.h. sie weisen das gleiche Konvergenzverhalten auch auf einer allgemeineren Klasse von Funktionen auf. Wir diskutieren verschiedene Verteilungen von zufälligen Suchrichtun-

gen und schätzen für jede Verteilung die erwartete Konvergenzrate ab.

Im zweiten Teil stellen wir drei Verfahren vor, die auf den ersten Blick keinen gemeinsamen Bezug aufweisen. Wie zeigen aber, dass sie alle zur Klasse der Zufallsjagdverfahren gehören und mit unserer Methode analysiert werden können. Die Beispiele umfassen (i) das Lösen von linearen Gleichungssystemen mit der Kaczmarz-Methode und (ii) das Schätzen einer Hesse-Matrix nach einer Methode von Leventhal und Lewis. Beide Verfahren sind spezielle Anwendungen von Suchverfahren mit exakter Liniensuche, deren Schritte durch die Werte von bestimmen Skalarprodukten eindeutig festgelegt sind und eine einfache geometrische Interpretation erlauben. Als letztes Beispiel (iii) analysieren wir dieses spezielle Verfahren in allgemeinen Hilberträumen und leiten exakte Konvergenzraten her. Das Verfahren zum Schätzen einer Hesse-Matrix hat auch eine weitere konkrete Anwendung: es kann verwendet werden um die intrinsische Metrik eines Optimierungsproblems zu schätzen. Wird diese Schätzung bei der Wahl der Suchrichtungen berücksichtig, kann dies die Konvergenz von Zufallsjagdverfahren beschleunigen. Wir zeigen, dass eine bestimmte Implementierung dieses zweistufigen Verfahrens auf allen quadratischen Funktionen gleich schnell konvergiert, d.h. dieses Verfahren ist affin invariant.

Im letzten Kapitel diskutieren wir Nesterovs gradientenbasierte Beschleunigungstechnik für zufallsgesteuerte Suchverfahren. Jede Iteration dieses Verfahrens besteht aus zwei Schritten: (i) einer einfachen Suche nach einem besseren Suchpunkt, wie in den Zufallsjagdverfahren, und (ii) der Aktualisierung einer Schätzung der Zielfunktion (Modell), welche die Grundlage bildet für die schnellere Konvergenz. Wir zeigen, dass in Schritt (i) gefahrlos eine Liniensuche verwendet werden kann, aber wir können dies im allgemeinen Fall nicht auch für Schritt (ii) zeigen. Auf quadratischen Funktionen konvergiert das beschleunigte Verfahren mit Liniensuche mindestens gleich schnell wir die einfachen Zufallsjagdverfahren—möglicherweise aber deutlich schneller.

Acknowledgements

First and foremost, I would like to express my gratitude to both my advisers Christian Müller and Bernd Gärtner. I like to thank Christian for his excellent support at the beginning of my PhD, uncountable many encouraging discussions and his hospitality during my visit in New York; and Bernd for his continuous support, encouragement, and guidance when needed most. Without both of you, this would not not have been possible!

My thanks go to Emo Welzl for letting me be part of his research group and providing such a perfect working environment, and to Ivo Sbalzarini for taking me on in the MOSAIC group in the early days. Many group meetings and intensive discussions allowed me to gain diversified insight in many interesting applications. I would like to thank Jonathan Goodman for inviting me to visit the Courant Institute in New York.

My sincere thanks go to Yurii Nesterov for accepting to review my thesis and his helpful remarks.

I gratefully acknowledge the funding received from the Computational Geometric Learning (CGL) project. CGL was funded by the Future and Emerging Technologies unit of the European Commission (EC) within the 7th Framework Programme for Research of the EC, under contract No. 255827.

I would like to thank all the current and former GREMOs whom I had the pleasure to work with: Yves Brise, Tobias Christ, Andrea Francke, Heidi Gebauer, Anna Gundert, Timon Hertli, Michael Hoffmann, Martin Jaggi, Vincent Kusters, Robin Moser, Gabriel Nivasch, Andrea Salow, Dominik Scheder, Marek Sulovský, May Szedlák, Antonis Thomas, Hemant Tyagi, Uli Wagner, and Manuel Wettstein. I would also like to thank all the members of the MOSAIC group, especially Omar Awile and Janick Cardinale for their help and hospitality, Grégory Paul, Rajesh Ramaswamy and Sylvain Reboux for their expertise.

I like to thank all other colleagues whom I had the pleasure to meet at a conference or workshop—or much simpler: on the same floor in CAB—and whom I had the opportunity to get to know better. In particular, I am also grateful to my office mates: long-term companion Timon, Manuel, and the long-term guests Zuzana Safernová and Arnau Padrol. I also shared the pleasure to work together with Martin and Hemant on some Machine Learning problems that did not find their way into this thesis.

I am grateful to my family; to my girlfriend Eva; and to my best friends for all their support and the great time. In particular, I like to mention my frequent teammates Daniel and Marco for not getting desperate when loosing once more against Christian and "Schiltenpulli" Reto; as well as Avanti, Buzz, Frostie, Geno, Hathi, Idefix, Onari, Nilsson, Piano, and Zippo for the epic battles of elements and all other adventures.

Contents

Chapter 1

Introduction

In this thesis we study the unconstrained optimization problem

$$f^* := \inf_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \tag{OPT}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is a continuous function on an n-dimensional vector space over $\mathbb{R}$. We are given *oracle access* to f, that is we can query the value of f at any point $\mathbf{x} \in \mathbb{R}^n$. Our goal is twofold: by querying the values of f at multiple points we (i) aim to find an *approximate solution* $\mathbf{x}' \in \mathbb{R}^n$, that is a point $\mathbf{x}'$ whose function value $f(\mathbf{x}')$ is close to f^*, and (ii) the number of function evaluations should be as small as possible. For problem (OPT) the minimum need not be attained. Therefore we will either have to enforce the existence of a minimum by additional constraints, for instance by restricting the minimization to a bounded domain, or, and this is what we do in the remainder of this chapter, simply assume that at least one point $\mathbf{x}^* \in \mathbb{R}^n$ with $f(\mathbf{x}^*) = f^*$ exists.

1.1 Black-Box Optimization

In the optimization problem (OPT) the objective function f is "hidden": we don't have any *a priori* knowledge about f but we can observe its values $f(\mathbf{x})$ for any point $\mathbf{x} \in \mathbb{R}^n$ of our choice. It behaves like a "black box" and we thus refer to problem (OPT) as a *black-box optimization problem*.

One might wonder where we actually encounter such black-box functions. One unlimited source of black-box problems is obviously nature

itself. The functions could for instance measure energies (or "qualities") of some physical particle, chemical system or biological structure. However, we are often not interested in only finding extremal values of such functions, but science is all about understanding the *structure* behind it. In its purest form, the structure can be captured by a simple formula—a *law* in physics—but in most applications from biology, chemistry or physics one has to settle for a partial description of the energy function (see e.g. [166] and references therein).

The optimization problem (OPT) is more prominent in engineering. There the primary goal is not to study the structure of the function, but to find the best (or a suitably good) solution. For a typical engineering task it is certainly plausible to assume that we can indeed evaluate the objective function f for a large number of search points, in contrast to the black-box functions from nature. Thus we can design algorithms that try to find an approximate solution to problem (OPT) by repeatedly evaluating the function f at various inputs points.

Before studying the design of specific schemes we must emphasize at this point that there cannot be *one* universal algorithm that approximates all optimization problems (OPT) in reasonable time. This follows for instance from the "no free lunch theorem" of Wolpert and Macready [218, 256, 257]. Thus, for every algorithmic scheme one should also specify a certain *class* of problems, i.e. a set of functions with well-specified properties, on which the scheme can (i.e. is intended to) be applied.

1.1.1 Convex Optimization

Some of the most important problem classes are *convex* problems. Convex functions exhibit a very strong global structure: for every segment, the value of the function at the segment midpoint does not exceed the mean of the values at the end of the segment.[1] This property especially implies that every *local solution*[2] is also a *global solution* of problem (OPT).

The development of optimization algorithms that only use function values dates back to the 1950's [33, 37]. However, the interest in these simple schemes dropped rapidly in the 1970's. The reason is simple: the special *structure* of convex problems allows for very efficient optimiza-

[1] A function $f\colon \mathbb{R}^n \to \mathbb{R}$ is convex if $f\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right) \leq \frac{f(\mathbf{x})+f(\mathbf{y})}{2}$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. This is equivalent to $f(t\mathbf{x}+(1-t)\mathbf{y}) \leq tf(\mathbf{x})+(1-t)f(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and $t \in [0,1]$.

[2] A point $\mathbf{y} \in \mathbb{R}^n$ such that there is an $\epsilon > 0$ with $f(\mathbf{y}) \leq f(\mathbf{x})$ for all $\|\mathbf{x}-\mathbf{y}\| \leq \epsilon$.

tion algorithms if in addition to the function values also gradient vectors (or subgradients for non-smooth functions) are available [178]. This assumption is well-justified for many applications: the development of the fast differentiation technique in the 1980's showed that if the objective function is explicitly given by a sequence of differentiable operations (say, a computer program that computes the function values) one can also write down an (efficient) program for computing the whole vector of its partial derivatives [72, 131, 206]. Nowadays, the theory of convex optimization is well-understood [34, 178, 181, 197] but still an active field of research. We will review a few important results in Section 2.1.

From an efficiency point of view it makes in general no sense to neglect the gradient information if it is available [44, 184, 239]. However, it was noted in [44] that "computing the derivative is the greatest single source of errors" in many programs, thus it can make sense to trade computation-time versus the time it takes to write a program that computes the gradient. Among the algorithms that only use function values we distinguish two classes: *gradient-based* algorithms (see e.g. [184]) that rely on an approximation of gradient information by e.g. finite-differences, and *gradient-free* or *direct* algorithms (see e.g. [106, 175, 243]) that neither compute nor approximate the gradient explicitly and can even be applied on problems where no gradients exist.

1.1.2 Outside the Box

We now leave the convex world and risk a glimpse on non-convex optimization problems. Gradient-based schemes struggle on functions with many local optima, as they get often stuck in local minima and fail to find a global solution. However, gradient-free schemes can show good performance when applied to practical problems—at least empirically. But strong theoretical results have typically still not been attained.

Simple schemes for non-convex problems have been around since the early days (see e.g. [159, 169, 207, 208, 220]). The advances of modern computers made it possible to implement more advanced schemes and attack various interesting problems. Starting in the 1990's, these includes applications in engineering design, circuit design, medical image registration, dynamic pricing (see [44]), molecular geometry (see [166]), error analysis of Gaussian elimination [103], parameter tuning of nonlinear optimization methods [11], quantum control [35], and parameter estimation in systems biology networks [241].

Just to name one exemplary scheme, we would like to point out the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [89, 93–95]. This scheme shows excellent performance on benchmark problems [92, 148, 149], and has proven its efficiency also in several applications (see e.g. [90]). Together with highly related schemes it forms the state of the art in the field.

Above we have seen that there is a vast area of black-box optimization problems that have been successfully tackled with schemes that only query function values, but theory is lagging behind. However, there is no clear mathematical description of the function classes on which such schemes perform well. Practitioners sometimes claim—or observe—that their schemes work well for so-called "real-world problems". This obscure term does certainly not comprise *all* real "real-world" problems, only the ones that can nowadays be handled to certain satisfaction. To give the reader a rough picture, we would like to emphasize that functions which are "close" to a convex function on a global scale (but with maybe many local minima), and mixtures (convex combinations) or noisy versions of such functions can be considered tractable to some extent.

There have been advances towards descriptions of non-convex function classes [52, 109, 193, 198], or descriptions of complicated functions by summary statistics [164, 166, 168]. Popular benchmark suites (cf. [92, 148]) assess the performance of such heuristics by comparing their performance on "typical" problems. Thus from a far fetched view, those examples could give a definition of what is considered to be a "real-world problem". Whilst there are certainly a lot of gaps to fill in this area, we did not step into this direction.

1.1.3 Towards Theory

We have already mentioned that for any specific scheme, we can only expect strong convergence results to hold for a subclass of problems. Hence, it is desirable that this class at least comprises some easy problems, e.g. convex or sometimes also *quasi-convex* (functions with convex level sets) problems. Therefore, mostly due to lack of alternatives, such algorithmic schemes are studied *not* on the problem class that they have been *designed for*, but only on convex functions. These theoretical results serve the purpose to show that the schemes do not just accidentally work in rare circumstances, but that they actually are able to find the minimum of simple convex functions with high probability. The results

can be classified into two types: (i) pure convergence results, that show that the schemes do not prematurely converge to suboptimal points or even diverge, but typically these proofs come without concise bounds on the running time. The second type is more practical: by proving (ii) upper bounds on the *convergence rate* one immediately gets bounds on the running time (to solve the problem up to a given accuracy), and one has a qualitative measure to compare the efficiency of different schemes. One should not forget that this approach can in general not measure the ability of the schemes to generalize to non-convex problems.

There has been a lot of progress in this area lately (see e.g. [44]), but let us just mention some exemplary results related to the aforementioned CMA-ES. Some early predecessors of this scheme [112, 169, 207, 220] have been studied on very simple *example problems*, like for instance linear functions or quadratic functions of the form $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x}$. Setting $A = I_n$, the n-dimensional identity matrix, yields the "sphere" function, other popular examples comprise of matrices with only two different eigenvalues: the smallest appearing with multiplicity one ("cigar"), the largest with multiplicity one ("discus" or "tablet") or both with equal multiplicities ("two-axes") (see e.g. [93, 95]). This approach has also been used for more advanced schemes (see e.g. [25, 27]). However, the arguments are often very "geometric" and don't allow for easy generalization to arbitrary convex functions. Some pure convergence results (without rates) have for instance been recently derived in [4, 12, 13, 116, 117]. In the past few years—almost in parallel to this thesis—a promising approach has become popular: the framework of "information-geometric optimization" tries to explain schemes like CMA-ES [5, 26, 191], or variants termed Natural Evolution Strategies (NES) from a more abstract point of view [20, 75, 219, 252, 253, 260]. In a nutshell, these schemes can be interpreted as a version of Gradient Descent on an abstract space of probability distributions, see e.g. [26, 252]. This approach is very appealing, as it treats large classes of objective functions. However, exact convergence rates are still not known.

1.2 Random Pursuit Framework

We started out to find a theoretical framework that would allow to derive convergence rates for variants of CMA-ES or similar schemes. CMA-ES is an iterative algorithm that works roughly speaking as follows. At each iteration (i) it *explores* the neighborhood of a search point by sampling

trial points from a probability distribution and evaluating the objective function at these points; (ii) a new search point is generated, for instance simply as a weighted average of the trial points; and (iii) a new sampling distribution is generated, for instance by updating the old distribution and incorporating some acquired knowledge of the objective function. We will provide a slightly more detailed description of this scheme later in Chapter 2 below. Now we present the contents of this thesis.

We study a certain class of schemes for problem (OPT) that only query function values. We assume that the schemes iteratively generate a sequence of search points. Each iteration comprises two simple elementary primitives: (i) generation of a *search direction* and (ii) generation of a *step*, that is, picking the next search point from a line that is defined by the old search point and the search direction. For a search point $\mathbf{x} \in \mathbb{R}^n$, a search direction $\mathbf{u} \in \mathbb{R}^n$ and a step size $\sigma \in \mathbb{R}$, we can express one step of such a scheme as

$$\mathbf{x}_+ = \mathbf{x} + \sigma \cdot \mathbf{u}\,,$$

where $\mathbf{x}_+ \in \mathbb{R}^n$ denotes the next iterate that is reached after this step. We refer to schemes that can be cast into this framework as *Random Pursuit* algorithms. The name avers that the search direction in (i) can be generated by simply drawing a random sample from a probability distribution. For instance, the uniform distribution on the unit sphere determines an unbiased "pure" random scheme, whereas a Dirac distribution is equivalent to a deterministic scheme. We measure the quality of the new search point $\mathbf{x}_+$ by means of a *sufficient decrease condition* and derive convergence rates on smooth convex functions. The derived rates depend on the sufficient decrease condition and on the *full* eigenvalue spectrum of the objective function. For example, for the quadratic function $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x}$ from above, these are the eigenvalues of the matrix A. This extends the classical literature [34, 182] which often only considers the extremal eigenvalues.

Due to the generality of the assumptions, many existing black-box optimization algorithms can be described in terms of the above two primitives and therefore fit in our framework. For instance, we consider examples where the directions $\mathbf{u}$ are sampled from the surface of a hypersphere (or -ellipsoid), and the steps σ are generated by gradient-free (pure theoretical) exact line search or a (more practical) inexact line search procedure. These sampling distributions are exactly the same as used in CMA-ES and, as we will argue, the steps of CMA-ES can be regarded as inexact line searches. Considering an exact line search

oracle has the following benefit: the generated steps σ are independent of the exact shape of the objective function along the search direction. For the one-dimensional functions $|x|^2$, $|x|$ and even $|x|^{1/2}$, an exact line search oracle will always identify 0 as the unique optimum. This allows generalization of the results to broader function classes.

We derive convergence rates for various *fixed* sampling distributions for search directions $\mathbf{u}$. Many successful optimization schemes, including CMA-ES, comprise a mechanism to adapt the sampling distribution in order to fit better to the optimization problem (OPT); allowing faster convergence. We address this topic by reviewing a mechanism from Leventhal and Lewis [141]. This scheme is different from the one in CMA-ES, but it only uses function values, too. However, it cannot be termed gradient-free as it relies on finite differences. This scheme not only converges to the optimal sampling distribution on convex functions, it is also interesting from a second point of view: it can be analyzed in the Random Pursuit framework. The scheme samples in each iteration search directions from a (simple) distribution on the unit sphere and uses exact line search to generate the steps.

Leventhal and Lewis [141] use their scheme in the setting of optimization: when facing an optimization problem (OPT) one should first use this scheme to estimate a good sampling distribution, and then use a Random Pursuit algorithm which samples from this estimated distribution. If the problem requires it, one could also repeat these two steps. By changing between estimation and search in every iteration, we obtain an optimization algorithm that is conceptually similar to CMA-ES and amenable to theoretical investigation.

As a further application, we remark that the problem of learning the optimal sampling distribution is related to solving a linear system of equations and illustrate that the randomized Kaczmarz method [122, 240] is also a Random Pursuit algorithm, even with exact line search. We see that although the assumptions above—especially the exact line search—sound restrictive at first sight, Random Pursuit schemes can have many applications. However, they might only reveal themselves after a closer look.

As a last topic, we investigate whether it is possible to accelerate the Random Pursuit algorithms. For this, we have to consider schemes whose iterations comprise steps that are slightly more involved than the two primitives mentioned above. Recently, Nesterov [184, 185] has shown that gradient-based randomized schemes can be substantially accelerated. We investigate if gradient-free schemes, especially line search

based methods, can attain the same optimal rates.

1.2.1 Previous Work

As a start, we studied a specific example of a Random Pursuit algorithm [239]. A scheme that samples search directions uniformly from the unit sphere and determines the steps by a line search. The idea to use an (exact) line search was inspired by a paper of Kleiner, Rahimi and Jordan [134]. These authors present an algorithm called Random Conic Pursuit (RCP) which accesses in each iteration a *two*-dimensional optimization (or line search) oracle. This oracle makes the proofs quite elegant and thus we tried to follow this idea.

Random Pursuit with isotropic sampling distribution, i.e. the uniform distribution on the unit sphere, has already been discussed in the literature, it appears in the work of Mutseniyeks and Rastrigin [169] and was analyzed later by Karmanov [126, 127, 264]. The decision to present the results in this thesis with respect to the sufficient decrease condition (instead of more direct assumptions on the line search, as we did in [239]) was inspired by the presentation in [264]. We enhance Karmanov's results in a number of ways: (i) we prove expected convergence rates also under certain versions of *approximate* line search; and (ii) for much more general sampling distributions.

As already mentioned, the scheme for learning an optimal sampling distribution is due to Leventhal and Lewis [141]. The accelerated scheme for gradient-based methods is due to Nesterov [184, 185] and we were also influenced by a recent presentation by Lee and Sidford [140].

1.3 Contents and Contributions

Chapter 2. We provide background information to familiarize non-expert readers with (convex) optimization. We present a review of complexity results and important and popular algorithms, especially present some schemes related to CMA-ES. We discuss general challenges that algorithmic schemes are facing, allowing to appreciate the abstractions and simplifications that were taken in the Random Pursuit framework.
Chapter 3. We provide convergence results for Random Pursuit algorithms that respect a sufficient decrease condition. The results apply to deterministic and randomized schemes. We strengthen the general results for Random Pursuit algorithms that explicitly use a line search.

This chapter is based on [237, 239].

Chapter 4. We detail specific algorithms that fit into the Random Pursuit framework. We discuss exact and inexact line searches that generate the steps and a variety of sampling distributions to generate the search directions.

The examples are mostly extracted from [237, 239], with a few new additions.

Chapter 5. We present applications of three (almost identical) Random Pursuit algorithms. We revisit a recent scheme of Leventhal and Lewis [141] that can be used to estimate an optimal sampling distribution for Random Pursuit algorithms. We show that the error bounds from [141] are optimal up to a factor of 2 and present an implementation of this scheme that is independent of the initial approximation error. Hence, an algorithm that uses this technique to estimate the sampling distribution converges equally fast on all quadratic functions, i.e. is affine invariant. The last example comprises Kaczmarz' method [122, 240] for solving systems of linear equations.

This chapter is based on [237], the technical reports [234, 238] and motivated by empirical data from [235].

Chapter 6. We discuss an acceleration technique for simple Random Pursuit algorithms. We review results for gradient-based schemes and provide preliminary results for truly gradient-free, line search based schemes on quadratic functions.

This chapter summarizes so far unpublished ongoing work and ideas. The study of accelerated schemes for line search algorithms was motivated by promising empirical data reported in [233, 236, 239].

Chapter 7. We conclude this thesis in Chapter 7.

Chapter 2

Background

Here, we present some fundamental background material for convex optimization. In the first two Sections 2.1 and 2.2 we present (classical) optimization algorithms and complexity results; Section 2.4 is devoted to Evolution Strategies. In Section 2.3 we discuss the components of algorithmic schemes in more detail. In particular, we comment on the search directions and step sizes, and how we measure their quality.

2.1 Methods for Optimization

We give an overview of the most fundamental optimization algorithms. We present the cornerstones of the field and focus on schemes that are in some way related to the work in this thesis. That is, we emphasize especially schemes that do not rely on derivatives, and neglect the numerous advances in various areas of gradient-based optimization.

We structure our listing by function classes that can from a theoretical point of view be addressed with the specific schemes. However, sometimes this classification is unclear or yet undecided, thus we will subsume most of the derivative-free search "heuristics" in the last Section 2.1.3.

To avoid technicalities at the moment, we consider throughout this section the *constrained* problem

$$f^* := \min_{\mathbf{x} \in \mathbb{R}^n \cap B_{\mathbf{0}}(R)} f(\mathbf{x}) \,, \qquad (\mathrm{OPT_R})$$

where $B_{\mathbf{0}}(R)$ is the ball of radius R around the origin. It is clear that

when (OPT) has a solution $\mathbf{x}^* \in \mathbb{R}^n$, it will be contained in $B_{\mathbf{0}}(R)$ for R large enough.

2.1.1 Global Optimization

There is no algorithm that efficiently computes an approximate solution for *any* arbitrary objective function [218, 256, 257] (see also Section 2.1.1 below). Still, we would like to point out two popular approaches that can be applied to arbitrary objective functions—though one might have to wait forever to approximately solve problem ($\mathrm{OPT_R}$). Both are related to sampling. Generating random samples from $B_{\mathbf{0}}(R)$, picking each point $\mathbf{x} \in B_{\mathbf{0}}(R)$ with probability proportional to $\max\{c - f(\mathbf{x}), 0\}$, for some (arbitrary) normalization $c = f(0)$, allows to identify the neighbourhood of a global optimum—if the function behaves nicely. A related approach is to iteratively explore the search space by generating samples only in the neighborhood of the current iterate and move to points with better function values. At first, to allow exploration of the whole space, also points with worse function values are accepted with a specified acceptance probability. Over time, the acceptance probability is decreased in such a way that the algorithm converges to a local optimum. Those two ideas can be turned into algorithmic schemes. The first one is known as the Metropolis-Hastings (Markov chain Monte Carlo) algorithm [97, 162], the second one as Simulated Annealing [132, 247]. Asymptotic convergence results to a global optimum have been presented but there is no guarantee that a good solution will be obtained in a finite number of iterations [211]. Interesting finite-time performance aspects are discussed in [41, 192].

Those algorithms can be sped up when using an appropriate prior distribution that encodes some knowledge about the function. Alternatively, promising regions could also be learned during optimization and the sampling distribution accordingly adapted [6, 87, 146]. Variants of these heuristics are nowadays applied in various fields of research [21, 46, 73, 88, 244].

Lipschitzian Optimization

One of the nice properties of the objective function could for instance be Lipschitz continuity[1]: function values can only change continuously and not too fast. Thus, points in a small neighborhood of the global

[1] A function $f\colon \mathbb{R}^n \to \mathbb{R}$ with $\frac{|f(\mathbf{x})-f(\mathbf{y})|}{\|\mathbf{x}-\mathbf{y}\|} \leq L$ for an $L \geq 0$ and all $\mathbf{x}, \mathbf{y} \in B_{\mathbf{0}}(R)$.

optimizer will be approximate solutions of problem ($\mathrm{OPT_R}$). However, the expected time to find such a solution by either random sampling or exploring the space along predefined meshes is still exponential in the dimension (see e.g. [182]).

Other schemes make use of the Lipschitz property in a more clever way: they determine lower bounds on the function values inside a hyperrectangle if the function values at its corners are known. Thus if already a better search point has been found, this hyperrectangle can be excluded from the search [119, 227]. However, it is not hard to see that all those schemes still need exponential (in the dimension) time to find a search point in the neighborhood of the global optimum [36].

2.1.2 Convex Optimization with Derivatives

Now we discuss convex objective functions. Each convex function on $\mathbb{R}^n$ is necessarily continuous, but not necessarily differentiable. Smooth functions can be linearly approximated at any point $\mathbf{x} \in \mathbb{R}^n$ with first-order Taylor expansion, i.e. a linear function passing through $f(\mathbf{x})$ at $\mathbf{x}$, whose slope is given by the gradient $\nabla f(\mathbf{x})$ (vector of first-order derivatives). By convexity, this linear function *underestimates* the convex function everywhere. If f is not differentiable at $\mathbf{x}$, the linear lower bound is not unique anymore. The set of slopes of all linear lower bounds that pass through $f(\mathbf{x})$ at $\mathbf{x}$ is called the *subdifferential*[2], and its elements *subgradients*.

Nonsmooth Functions

The subgradient is a *separation oracle*: it allows to identify regions of the search space, where the global optimizer cannot be. The Center of Gravity method [142, 189] makes use of this very explicitly. This method *localizes* all of the minimizers $\mathbf{x}^*$ of problem ($\mathrm{OPT_R}$) it the following way: starting from the bounded domain $Q_0 = B_{\mathbf{0}}(R)$, it computes in each iteration k a halfspace H_k that does contain $\mathbf{x}^*$ and has the center of gravity of Q_k on its boundary. Then set $Q_{k+1} = (Q_k \cap H_k)$. By Grünbaum's inequality[3] [85], the volume of the feasible sets decreases by a constant factor in each iteration, and therefore only a linear number of iterations are necessary to decrease the volume by an exponential factor. However, we must note that this method is not practical, as

[2] Formally, $\partial f(\mathbf{x}) := \{\mathbf{v} \colon f(\mathbf{y}) - f(\mathbf{x}) \geq \langle \mathbf{v}, \mathbf{y} - \mathbf{x} \rangle \ , \forall \mathbf{y} \in \mathbb{R}^n\}$.

[3] For convex set C and halfspace H containing the c. of gravity: $\frac{\mathrm{vol}(C \cap H)}{\mathrm{vol}(C)} \geq \frac{1}{e}$.

the computation of the center of gravity of a convex set is a computationally hard problem, even for polytopes [205]. There are at least two ways to overcome this issue: the classical approach is the Ellipsoid method [29, 84, 225, 246, 262, 263] which introduces the following trade-off: instead of working with arbitrary convex sets Q_k it relies on ellipsoids P_k. The intersection of a halfspace with an ellipsoid is not any more an ellipsoid, thus in every iteration an ellipsoid P_{k+1} has to be computed that contains $(P_k \cap H_k)$. This can be done efficiently with $O(n^2)$ simple arithmetic operations, but the volume of the ellipsoids decreases only by a factor of roughly $\left(1 - \frac{1}{2n}\right)$ each time and a linear number of iterations are required to reduce the volume of the feasible set by a constant factor. Hence, this scheme requires $\Theta(n)$ times more iterations than the center of gravity method. A different approach is to use only *approximations* of the center of gravity, for instance obtained by Monte Carlo integration. This line of research is stimulated by the breakthrough result of Dyer et al. [55, 150, 152] who showed that random samples of a convex body can be generated in polynomial time.

Smooth Functions

Smooth optimization[4] makes use of the fact that the negative gradient points always to regions with better function values. Following the (negative) gradient direction leads to the minimizer of $(\mathrm{OPT}_\mathrm{R})$. This idea dates back to Cauchy [40] and is the base of the Gradient Descent method (see e.g. [34, 178, 181, 197]). Although very popular, Gradient Descent is not the most efficient method. We will elaborate on this in Section 2.3.3. Optimal schemes are Nesterov's Fast Gradient method [179, 180, 182, 186], Powell's Heavy Ball method [197], and variants thereof [16, 17, 245].

So far, these methods use only gradient information. The *Hessian* matrix (i.e. second derivative) defines a metric that describes the local (quadratic) structure of the objective function. Methods that make use of this information can converge much faster (see e.g. [190]). Important examples are the Conjugate Gradient algorithm [61, 102] which belongs to the general class of Krylov subspace methods (see e.g. [228]), and the general Newton-Raphson scheme (see e.g. [190, 261])indexNewton-Raphson scheme. Such schemes are termed *variable metric* methods. Despite this speed-up, there are two main disadvantages: (i) the second-

[4] In this thesis we mostly consider continuously differentiable functions and sometimes also twice continuously differentiable functions.

order information is rarely available, and (ii) most crucially: this amount of information (typically an $n \times n$ matrix) has to be processed and stored. For instance, for one iteration of Newton's method, the inverse of the Hessian matrix has to be computed (typically implemented by solving a system of linear equations). This prohibits their use in nowadays big-scale data processing tasks.

Quasi-Newton methods (see e.g. [190]) are a special case of variable metric methods that do not compute the Hessian directly, but instead maintain and update an approximation to the Hessian only using gradient vectors. The first quasi-Newton method has been proposed in the late 1950's by Davidon [48]. The BFGS method [38, 60, 78, 224] and its low-memory extension L-BFGS [145] belongs among others [19] to the most popular schemes (see e.g. [190]). Instead of approximating the Hessian, those schemes typically directly compute an approximation of the inverse Hessian matrix, avoiding the necessity to solve a linear system. Quasi-Newton methods are ubiquitous in all areas of science and engineering.

2.1.3 Convex Optimization without Derivatives

In the following, we give a short overview of some important methods for derivative-free optimization; for further reference we direct the reader to the comprehensive surveys [25, 128, 136, 143, 202, 210, 222, 259]. Many of the following schemes are *randomized algorithms*, that is they use in addition to the observed function values some internal randomness to guide the search. It was already recognized in the 1950's that randomization is one of the keys for successful derivative-free optimization [37].

Gradient-Based Methods

Gradients can either be estimated directly or indirectly. The latter methods estimate a *model* of the objective function and use its gradient. Box and Wilson [33] described such a method already in 1951. Typical schemes of this kind are Trust-Region methods. They use a surrogate model that is usually smooth, easy to evaluate, and presumed to be accurate in a neighborhood (trust region) of the current iterate. Powell [199] first proposed to use a linear model of the objective within a Trust-Region method, this was later also extended to quadratic models in [200, 201] or by Conn et al. [43]. For other extensions see e.g. [118].

In addition to, or instead of developing a surrogate of the objective function, one may develop an estimate of the gradient and use it to ex-

pedite the search. Implicit Filtering [74] uses an approximation of the gradient to guide the search, resembling the Gradient Descent method when the gradient is known. Nesterov [184] presented a method that moves along random directions; the step sizes are proportional to directional derivatives (estimated by finite differences with fixed increment). A similar approach is taken in [54]. Gradient-based methods are reported to be not very reliable if the objective function is noisy or has many local minima [136].

Gradient-Free Methods

A first class of gradient-free algorithms uses the localization approach from Section 2.1.2. For instance Protasov [203] mimics the behavior of the Ellipsoid method in terms of convex cones, and Bertsimas and Vempala [22] the behavior of the Center of Gravity method by approximating the center of gravity with random samples. Somewhat related are sampling-based ideas as encountered in Section 2.1.1. Ball-Walk [150] and Hit-and-Run [18, 30, 230] are two schemes (see also [248]) that can be used to generate a uniform distribution in a convex body, e.g. the level set of a convex function. Each iteration of Hit-and-Run is based on two random experiments: a line passing through the current iterate is sampled uniformly at random, and a step is generated by sampling uniformly a feasible point from this line. This scheme can be used for optimization by only accepting points with better function values.

Other schemes are more "direct" in the sense that they explore the function in a more local way, similar to Gradient Descent. For instance Hooke and Jeevis' Pattern-Search [50, 106, 243] from 1961 explores the search space by probing function values along predescribed directions ("patterns") for various step sizes. Only a few years later, Spendley et al. [231] and Nelder and Mead [175] presented their simplex-based algorithms. Those schemes maintain $n + 1$ search points arranged in a simplex and use reflecting and contracting steps to replace the worst search point by a better one. However, this can fail to converge [258]. Other schemes sample points from (adaptive) meshes [1, 2, 11] or similar partitions [120].

Another line of research resulted in Evolution Strategies (ES). Rastrigin [169, 207, 208] introduced the fixed step size random search that samples search directions uniformly from the unit sphere and uses constant step sizes. For example Matyas [159] and Schumer and Steiglitz [220] soon proposed schemes with variable step sizes. Schumer

and Steiglitz [220] performed a thorough theoretical investigation of their scheme on some selected quadratic functions. Their scheme is almost identical to the ES studied by Rechernberg [209] and Schwefel [221]. ES use bio-inspired operators to generate new search points. Typically, they not only use single points to describe iterates, but a set (or population) of points, and in each iteration many new search points are evaluated (offspring) and a new population is formed (selection). Hansen's Evolution Strategy with Covariance Matrix Adaptation (CMA-ES) [89, 93–95] is a state-of-the-art variant of this scheme that evaluates the fitness of a population by weighted ranking, comprises elaborate step-size control strategies and variable metric functionality. We will add to the discussion of ES in Section 2.4.1 below.

Among related schemes that we did not mention so far are more bio-inspired schemes, like Genetic Algorithms [77], Artificial Neural Networks [98], Tabu-Search [76], Particle Swarms [129] or the recent Natural Gradient Descent [75, 219, 252, 260] which seems amenable to theoretical investigations [5, 26, 252].

Variable Metric Methods

A variable metric upgrade goes along well with most schemes from both of the above mentioned classes—if a correct metric can be estimated. Although "directional adaptation" has been conjectured to be useful for randomized gradient-free schemes in the late 1960's [220, 232] the early literature on this topic is scarce and scattered across different communities. Important examples include the Gaussian Adaptation algorithm developed by Kjellström and Taxen [133, 167] in the context of analog circuit design and Marti's controlled random search schemes using concepts from optimal control [155]. Nowadays, most of the successful derivative-free schemes comprise a mechanism to adapt and learn an underlying metric of the objective function. Such schemes are in particular implemented in Trust-Region methods [44], Powell's model based optimizers [200, 201] and CMA-ES [89, 95].

Despite their great appeal in practice, many randomized gradient-free variable metric schemes often lack a thorough theoretical convergence analysis.

2.2 Complexity

In this section, we formally define what it means to solve problem (OPT) approximately and present a few key results regarding the *complexity* of convex optimization problems. Here, complexity measures the number of oracle calls, i.e. function evaluations or gradient computations, that are required to solve problems from certain classes. The term should not be confused with *computational complexity* that counts the simple arithmetic operations that are necessary if the algorithms are implemented on a computer, say.

2.2.1 Algorithmic Schemes and Solutions

In order to define the complexity of an optimization problem (OPT), we must agree on the amount of information an algorithm is allowed to query. We assume *oracle access* to the objective function f. A p-th order oracle is a function $\mathcal{O}$ that returns for every query point $\mathbf{x} \in \mathbb{R}^n$ the $(p+1)$-tuple $(f(\mathbf{x}), \nabla f(\mathbf{x}), \ldots, \nabla^p f(\mathbf{x}))$, if these derivatives exist. For instance for non-differentiable functions, as encountered in Section 2.1.2, a first-order oracle is defined to return an arbitrary subgradient instead.

For a given initial position $\mathbf{x}_0 \in \mathbb{R}^n$ (starting point) and a (given) algorithm, we denote by $(\mathbf{x}_k)_{k\geq 0}$ the sequence of iterates generated by the algorithm. That is, all the oracle values $(\mathcal{O}(\mathbf{x}_k))_{k\geq 0}$ will sequentially be made available to the algorithm, and iterate k can depend on $\mathbf{x}_0, \ldots, \mathbf{x}_{k-1}$ and their respective oracle values.

The algorithm has to find an approximate solution $\mathbf{x}' \in \mathbb{R}^n$. Given a constant $\epsilon > 0$, we measure the absolute error as

$$f(\mathbf{x}') - \min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \leq \epsilon\,. \tag{2.1}$$

If the function f is M-Lipschitz continuous, then we can estimate the initial error $\big(f(\mathbf{x}_0) - \min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})\big)$ by the Lipschitz parameter M and the radius of the initial level set, simply R in case of the bounded problem $(\mathrm{OPT}_\mathrm{R})$ (see e.g. [181]). Similarly for smooth functions with L-Lipschitz continuous gradients, see Section 2.5.4.

For $\epsilon \geq 0$, the *running time* $N_{\mathcal{A}}^f(\epsilon)$ of an algorithm $\mathcal{A}$ on an objective function f is the smallest integer N such that $\mathbf{x}' = \mathbf{x}_N$ satisfies (2.1). For randomized algorithms, the running time is defined likewise, requiring that (2.1) holds in expectation. Typically, we are not interested in the performance of an algorithm on a *specific* function, but on a certain *class* or *set* $\mathcal{F}$ of functions. The running time on a class of problems is the

smallest number of oracle calls such that (2.1) holds for *every* problem of this class, $N_{\mathcal{A}}^{\mathcal{F}}(\epsilon) = \sup_{f\in\mathcal{F}} N_{\mathcal{A}}^{f}(\epsilon)$. The *complexity* of a class $\mathcal{F}$ is the minimal number of function evaluations that *any* algorithm needs to solve *all* problems of the class, $N^{\mathcal{F}}(\epsilon) := \inf_{\mathcal{A}} N_{\mathcal{A}}^{\mathcal{F}}(\epsilon)$. To give an upper bound on the complexity, it is sufficient to present an algorithm that solves all problems $f \in \mathcal{F}$. For a lower bound, typically a problem instance has to be constructed that is hard for *all* algorithms—a so called *resisting oracle*.

2.2.2 Complexity of Convex Problems

The complexity of convex optimization problems with *first-order* oracles is well studied, slightly less is known for zeroth-order oracles. Although the focus of this thesis lies on gradient-free schemes, we review results for both types of oracles for comparison reasons. To avoid degenerate situations, we always consider the bounded problem ($\mathrm{OPT}_{\mathrm{R}}$). For a more detailed presentation we refer to the books of Nemirovski and Yudin [178], Nesterov [181] and the lecture notes [176].

First-Order Oracles

The most important complexity bounds for first-order optimization are summarized in Table 2.1 and will be mentioned in our discussion below.

Convex optimization. For M-Lipschitz continuous functions one distinguishes two regimes: (i) *high dimensional* problems with $n\epsilon^2 \geq \frac{1}{4}$ and complexity $\Theta(\epsilon^{-2})$, independent[5] of the dimension n; attained e.g. by the Subgradient method [184]. When (ii) $n\epsilon \leq \frac{1}{2}$ the problem is *low dimensional* with complexity $\Theta\big(n \ln \frac{1}{\epsilon}\big)$; attained e.g. by the Center of Gravity method [142, 189] which needs $O\big(n(\ln M + \ln R + \ln \frac{1}{\epsilon})\big)$ iterations. Here, see that the notation of complexity shall not be confused with *computational complexity*. The Ellipsoid method [225, 262, 263] in turn needs $O\big(n^2(\ln M + \ln R + \ln \frac{1}{\epsilon})\big)$ iterations, but each one can efficiently be implemented with $O(n^2)$ simple arithmetic operations. The Monte Carlo scheme [22] needs the same number of oracle calls, but each iteration (generating the random samples) has computational complexity $O(n^6)$. For $\frac{1}{\epsilon} \in [2\sqrt{n}, 2n]$ the complexity is known up to a logarithmic factor.

[5] The complexity of high dimensional constraint problems depends on the geometry of the feasible set [176].

problem class	dimension n vs. accuracy ϵ	lower bound	optimal method
M-Lipschitz	$n \geq \frac{1}{4\epsilon^2}$	$O\left(\frac{M^2R^2}{\epsilon^2}\right)$	Subgradient method
	$n \leq \frac{1}{2\epsilon}$	$O\left(n \ln \frac{MR}{\epsilon}\right)$	Center of Gravity method
L-Lipschitz gradients	$n^2 \geq \frac{LR^2}{\epsilon}$	$O\left(\frac{L^{1/2}R}{\epsilon^{1/2}}\right)$	Fast Gradient method
	$n \ll \frac{LR^2}{\epsilon}$	$O\left(n \ln \frac{LR^2}{\epsilon}\right)$	Center of Gravity method
κ-convex		$O\left(\kappa^{1/2} \ln \frac{LR^2}{\epsilon}\right)$	Fast Gradient method

Table 2.1: Complexity of first-order convex optimization. The lower complexity bound is reached by the optimal methods.

Smooth convex optimization. For f smooth with L-Lipschitz continuous gradients, the complexity in high dimensions, ($n^2\epsilon \geq LR^2$), is $\Theta(L^{1/2}R\epsilon^{-1/2})$; attained by e.g. the Fast Gradient Method but not by Gradient Descent which needs $\Theta\big(LR^2\frac{1}{\epsilon}\big)$ oracle calls. For low dimensions, $n^2\epsilon \ll LR^2$, the complexity is $\Theta\big(n(\ln L + \ln R + \ln \frac{1}{\epsilon})\big)$. Nemirovski [176] points out: "in fixed dimension, the advantages of smoothness can be exploited only when the required accuracy is not too high".

Strongly convex optimization. On smooth functions with condition number κ (see Section 2.5) the complexity is $\Theta\big(\kappa^{1/2}(\ln L + \ln R + \ln \frac{1}{\epsilon})\big)$; achieved e.g. by the Fast Gradient Method. The logarithmic dependency on ϵ is also achieved by the Gradient Descent, which needs $\Theta\big(\kappa(\ln L + \ln R + \ln \frac{1}{\epsilon})\big)$ oracle calls, scaling with κ instead of $\kappa^{1/2}$.

Zeroth-Order Oracles

The lower bounds on the complexity from Table 2.1 still apply here, as we consider a more restricted oracle. Running times of specific algorithmic schemes imply upper bounds that we will present below and are summarized in Table 2.2.

The complexity theory of zeroth-order optimization is not as well developed as for first-order schemes and not many non-trivial lower bounds are known. Jägersküpper [114] shows a lower bound of $O\big(n \ln \frac{1}{\epsilon}\big)$ for a specific method (equivalent to Random Pursuit). However, his bound does not reveal the dependence on the condition number κ or on the Lipschitz parameter L.

Convex optimization. Again we assume f to be M-Lipschitz continuous. If the dimension $n = 1$, then the problem $(\mathrm{OPT}_{\mathrm{R}})$ reduces to a *line search*. The localization scheme (or simply bisection-search) solves this problem with at most $O\big(\ln M + \ln R + \ln \frac{1}{\epsilon}\big)$ function evaluations.[6] The higher dimensional analogues of these methods are the gradient-free localization schemes. Focusing only on the dependency on ϵ and n, we can state the following bounds: Protasov's method [178, 203] needs $O\big(n^2 \ln n(\ln M + \ln R + \ln \frac{1}{\epsilon})\big)$ function evaluations; a factor of only $\ln n$ more oracle calls than the Ellipsoid method. Bertsimas and Vempala's Monte Carlo scheme [22] is worse as it requires $O\big(n^5(\ln M + \ln R + \ln \frac{1}{\epsilon})\big)$ function evaluations. Nesterov's Random Gradient Descent [184] needs $O(nM^2R^2\epsilon^{-2})$ function evaluations, if the oracle can compute *exact directional derivatives* or $O(n^2M^2R^2\epsilon^{-2})$ pure function evaluations. The dependency on ϵ is the same as for the Subgradient method, but the schemes need $O(n)$ or $O(n^2)$ iterations more, depending on the accuracy of the oracle.

Smooth convex optimization. For f smooth with L-Lipschitz continuous gradients, the running time of Random Gradient Descent [184] can be bounded by $O\big(nLR^2 \ln \frac{1}{\epsilon}\big)$, a factor of n more oracle calls than required by Gradient Descent. We will obtain qualitatively similar estimates for the Random Pursuit algorithms considered in this thesis.

Strongly convex optimization. On smooth functions with condition number κ (see Section 2.5). The one-dimensional problem can again be solved with a simple localization scheme (or bisection-search). Strong convexity can be used to estimate the initial error, yielding a bound of $O\big(\ln \kappa + \ln R + \ln \frac{1}{\epsilon}\big)$ function evaluations, see e.g. [239]. A bound of the same order was later also derived in [115], but extending the results to optimization with inexact oracles.

In dimension n, Random Gradient [184] only needs $O\big(n\kappa(\ln L + \ln R + \ln \frac{1}{\epsilon})\big)$ function evaluations, in analogy to Gradient Descent. The Fast (or Accelerated) Random Gradient derived in the same paper [184] improves this to $O\big(n\kappa^{1/2}(\ln L + \ln R + \ln \frac{1}{\epsilon})\big)$. We will discuss gradient-based and gradient-free types of acceleration of the simple Random Pursuit algorithm in Chapter 6.

[6] The best constant is obtained for the Fibonacci method—a scheme dividing the segments according to ratios of subsequent Fibonacci numbers [130].

problem class	oracle	upper bound	method
M-Lipschitz	$f(\mathbf{x})$	$O\left(n^2 \ln n \ln \frac{M^2R^2}{\epsilon}\right)$	Protasov
	$f(\mathbf{x})$	$O\left(n^2 \frac{M^2R^2}{\epsilon^2}\right)$	Random Gradient (RG)
	$\langle \nabla f(\mathbf{x}), \mathbf{u} \rangle$	$O\left(n^2 \frac{M^2R^2}{\epsilon^2}\right)$	
L-Lipschitz gradients	$f(\mathbf{x})$	$O\left(n \frac{LR^2}{\epsilon}\right)$	Random Gradient (RG) *Random Pursuit (RP)*
κ-convex	$f(\mathbf{x})$	$O\left(n\kappa \ln \frac{LR^2}{\epsilon}\right)$	Random Gradient (RG) *Random Pursuit (RP)*
	$f(\mathbf{x})$	$O\left(n\kappa^{1/2} \ln \frac{LR^2}{\epsilon}\right)$	Accelerated RG
	$\langle \nabla f(\mathbf{x}), \mathbf{u} \rangle$		*Accelerated RP (SARP)*

Table 2.2: Upper bounds on the complexity of zeroth-order convex optimization and methods that reach these bounds. Upper bounds for RP are presented in Chapters 3 and 4, for SARP in Chapter 6.

2.3 The Components of Search Schemes

The Random Pursuit algorithms as introduced in Section 1.2 generate a sequence $(\mathbf{x}_k)_{k\geq 0}$ of iterates if applied to an optimization problem (OPT). We can write this sequence as:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \underbrace{\sigma_k}_{\text{step size}} \cdot \underbrace{\mathbf{u}_k}_{\text{search direction}} \tag{2.2}$$

where $(\sigma_k)_{k\geq 0}$, with $\sigma_k \in \mathbb{R}$ is a sequence of *step sizes* and $(\mathbf{u}_k)_{k\geq 0}$, with $\mathbf{u}_k \in \mathbb{R}^n$ is a sequence of *search directions*. Thus $(\mathbf{x}_k)_{k\geq 0}$ is defined, or *generated*, by the initial iterate $\mathbf{x}_0$ and the sequences of search directions and step sizes. In Chapter 3 we study the convergence of the sequence of function values $(f(\mathbf{x}_k))_{k\geq 0}$. To this end, we introduce conditions that both the sequences of step sizes and search directions have to satisfy. Below, we motivate these conditions by some (rather trivial) introductory examples. We also discuss fundamental aspects of randomized and accelerated schemes and comment on the bounded optimization problem ($\mathrm{OPT_R}$).

2.3.1 Step Size

To study the convergence of the function values of a sequence $(\mathbf{x}_k)_{k\geq 0}$ we investigate the *one step progress*, that is the quantity $f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})$

for every $k \geq 0$. We would like to emphasize that a *simple decrease*, i.e. simply a *nonnegative* one step progress is not enough to ensure convergence to a minimum. We demonstrate this with an example: consider the function $f_1(x) = x^2$ and let $(x_k)_{k\geq 0}$ be a monotonically decreasing sequence with $x_0 = 1$ and $\lim_{k\to\infty} x_k = \frac{1}{2}$. This sequence suffers from too small steps. Likewise $((-1)^k x_k)_{k\geq 0}$ does not converge to 0; the steps are too large. The sequence $((\ln k)^{-1})_{k\geq 0}$ converges to 0, but very slow.

To overcome this issue, one has to guarantee that the one step progress is not too small. This is enforced by a *sufficient decrease* condition. For instance, if gradients are available then such conditions can be formulated in terms of the gradient. Well-known conditions are the Armijo-Goldstein [7, 79], and Wolfe [254, 255] conditions.

In derivative-free optimization, the gradient cannot be accessed and it cannot easily be checked if the aforementioned conditions are satisfied. Thus, we impose even stronger conditions, that can be verified without accessing the gradient, but in turn imply sufficient decrease. We show in Chapter 3 that we can use a line search to enforce sufficient decrease. A *line search oracle* `LS` is a function that provides an *exact* solution to the one-dimensional optimization problem (OPT). Whilst this is a purely theoretical construct, in practice a (zeroth-order) line search algorithm solves the problem ($\mathrm{OPT_R}$) *approximately*, as discussed in Section 2.2.2.

In some cases, sufficient decrease can also be obtained differently. We would like to point out an interesting approach. Consider the following scheme: given $\mathbf{x}_k$, chose $\mathbf{x}_{k+1}$ *uniformly* from the level set $\{\mathbf{x} \mid f(\mathbf{x}) \leq f(\mathbf{x}_k)\}$. However, if the function f is not as simple as $f_1(x) = x^2$, or high dimensional, it is not clear how this idea could be efficiently implemented in a black-box setting. A more promising local approach is the following: choose the step size σ_k such that for a random normal direction $\mathbf{u}_k \sim \mathcal{N}(\mathbf{0}, 1_n)$, say, the *success probability*

$$\mathbf{Pr}\big[f(\mathbf{x}_k + \sigma_k \mathbf{u}_k) \leq f(\mathbf{x}_k)\big] = c\,, \tag{2.3}$$

for a constant $0 < c < \frac{1}{2}$. Intuitively, this prevents the two extreme cases of too small or too large steps. The steps cannot be too small, as for continuous convex f, we have $\lim_{\sigma\to 0} \mathbf{Pr}[f(\mathbf{x} + \sigma\mathbf{u}) \leq f(\mathbf{x})] = \frac{1}{2}$; unless $\mathbf{x}$ is the optimum. Likewise, the success probability approaches zero for too large steps. Calculating the *exact* value of σ_k is typically not required to implement the idea (2.3) in an algorithmic scheme. What people typically do is to rely on crude approximations of the success probability [209, 220]. We come back to this in Section 2.4.1 below.

2.3.2 Search Directions

The second component in scheme (2.2) are the search directions $(\mathbf{u}_k)_{k\geq 0}$. Like bad steps, suboptimal search directions can hamper the convergence. Consider the 2-dimensional analogue of the function f_1, that is $f_2(\mathbf{x}) := x_1^2+x_2^2$, with gradient $\nabla f_2(\mathbf{x}) = 2\mathbf{x}$. In iteration k, the one step progress can only be positive if $\mathbf{u}_k$ is not orthogonal to $\nabla f(\mathbf{x}_k)$. Hence, a sequence of bad search directions $(\mathbf{u}_k)_{k\geq k_0}$ with $\langle \nabla f(\mathbf{x}_k), \mathbf{u}_k\rangle = 0$ and $\nabla f(\mathbf{x}_k) \neq \mathbf{0}$ for $k \geq k_0$, prevents convergence to the minimizer $\mathbf{x}^* = \mathbf{0}$ of f_2, even if for instance the step sizes are always optimally determined by a line search. Let θ_k denote the angle between the gradient direction $\nabla f(\mathbf{x}_k)$ and the search direction $\mathbf{u}_k$. For a parameter $c \geq 0$, the squared *angle condition*

$$\cos^2\theta_k = \frac{\langle \nabla f(\mathbf{x}_k), \mathbf{u}_k\rangle^2}{\|\nabla f(\mathbf{x}_k)\|^2 \, \|\mathbf{u}_k\|^2} \geq c\,, \tag{2.4}$$

provides a lower bound on $|\theta_k|$, and the aforementioned degeneracy is avoided if the scheme satisfies (2.4) in every step. For randomized schemes, we measure the expected value of (2.4). This condition can be enforced in different ways. Suppose we have a set of candidate search directions that span $\mathbb{R}^n$. Then directions can either be picked at random from this set, or one can deterministically identify and pick the best direction. The angle condition (2.4) holds for $c = \frac{1}{n}$ if $\mathbf{u}_k$ is picked uniformly at random from the set $\{\mathbf{e}_i \colon i = 1, \ldots, n\}$ of standard unit vectors (see Example 4.14 below). This implies that there exists an index i such that (2.4) also holds for $\mathbf{u}_k = \mathbf{e}_i$. This observation is for instance used in the Pattern-Search algorithms [50, 106, 243]. The variant termed Compass Search [243] probes exactly the unit vectors $\mathbf{e}_i$ in each iteration.

Although a lower bound on the angle condition (2.4) together with sufficient decrease is enough to guarantee convergence, it might not be the right condition to ensure *fast* convergence. This can happen, if the gradient direction is not the "best" search direction that we should follow. Consider a skewed version of the function f_2, namely $f_3(\mathbf{x}) := 100x_1^2 + x_2^2$. The *condition number*[7], denoted as κ, measures the sensitivity of the function value subject to small changes in the argument, here $\kappa = 100$. Consider $\mathbf{x}^* = \mathbf{0}$ with $f_3(\mathbf{x}^*) = 0$. If we move one unit in direction $\mathbf{e}_1$, we reach $f_3(\mathbf{e}_1) = 100$, as opposed to $f_3(\mathbf{e}_2) = 1$.

[7] For a general definition of the condition number see Section 2.5.2 below.

In Chapter 4 we show that for a Random Pursuit algorithm—with exact line search oracle, say—that picks a search direction uniformly at random from the unit sphere, the number of iterations to find an approximate solution scales proportionally to κ. This also matches our intuition: every level set of f_3 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. If we want to move faster along the x_2-direction, we have to sample directions (almost) parallel to the x_2-axis more often. Hence, we have to sample the search directions from a unit sphere in a *different* norm—one that actually fits to the function f_3. By sampling uniformly from the set $\{\mathbf{x}\colon 100x_1^2 + 1x_2^2 = 1\}$, the number of iterations to find an approximate solution becomes independent of κ. Note that this anisotropic sampling distribution is specifically tailored for the function f_3. On other functions (take simply $f_4(\mathbf{x}) := x_1^2 + 100x_2^2$) this distribution is bad; even worse than the uniform distribution we started with at the beginning. Without prior knowledge on the black-box optimization problem, anisotropic sampling makes no sense at all. We emphasize that algorithmic schemes for problem (OPT) should therefore *adapt* their sampling distributions to the objective function.

Algorithms that adapt their behavior to the underlying norm, or *metric*, of the optimization problem (OPT) are referred to as *variable metric* schemes. If the objective function f is twice differentiable (e.g. as f_3 from above), a good metric is given by the second derivative, or *Hessian matrix*, at a point close to the optimum. We already have encountered several variable-metric schemes in Sections 2.1.2 and 2.1.3.

The left panel of Figure 2.1 depicts a Random Pursuit algorithm with a line search oracle $\texttt{LS}^f$. The line search on line 5 could alternatively also be replaced by any other method that guarantees sufficient decrease as discussed previously in Section 2.3.1. If the scheme is equipped with a routine to update the sampling distribution π, for instance by means of estimating the Hessian matrix or its inverse, then the sampling distribution can change either in every iteration or whenever some prescribed criteria are met.

2.3.3 Accelerated Schemes

A big handicap of variable metric schemes is the (obvious) fact that the correct metric is not automatically provided by the black-box optimization problem (OPT). Especially, if the metric has to be estimated

RP($f, \mathbf{x}_0, \pi, N$)
(schematic RP with line search and w/o variable metric)

```
if variable metric then
    π ← initialize(f, x_0)
for k = 0 to N − 1 do
    u_k ∼ π
    x_{k+1} ← LS^f(x_k, u_k)
    if variable metric then
        π ← update(π, f, x_k, x_{k+1}, u_k)
return x_N
```

ES($f, \mathbf{x}_0, \pi, N, \sigma, a, b$)
(schematic (1+1)-ES w variable metric)

```
for k = 0 to N − 1 do
    repeat
        u_k ∼ π
        if f(x_k + σu_k) ≤ f(x_k) then
            x_{k+1} ← x_k + σu_k; σ ← σ · a
        else x_{k+1} ← x_k; σ ← σ · b
        π ← update(π, f, x_k, σ, u_k)
    until x_{k+1} ≠ x_k
return x_N
```

Figure 2.1: Random Pursuit with line search $\texttt{LS}^f$ (left panel) and a (1+1)-ES with adaptive step size adaptation (right panel). In the variable metric version of RP, the sampling distribution π is updated in the beginning and after every or every few iterations. A specific choice for the routines initialize and update is discussed in Section 5.2; common updates used for ES are mentioned in Section 2.4.2.

solely by zeroth-order information it is not *a priori* clear whether one should invest these function evaluations to learn the metric or directly use them to guide the search towards the optimum. We will come back to this question later in Section 5.2.1. If the metric is represented by an $n \times n$ dimensional matrix, then estimating every single entry of this matrix might also not be very efficient from a computational point of view. Many schemes, like for instance the (first-order) L-BGFS [145] try to represent an approximation of the right metric with linear space, for instance as a combination of a low rank and a sparse matrix.

Now we present a completely different approach that also aims at accelerating the convergence rate of simple (random) search schemes. We observed in Section 2.2.2 that Gradient Descent needs $O\big(\kappa \ln \frac{1}{\epsilon}\big)$ first-order oracle calls to find an approximate solution to a strongly convex problem ($\mathrm{OPT_R}$) with condition number κ. In 1983, Nesterov developed Fast Gradient method [179] which needs only $O\big(\kappa^{1/2} \ln \frac{1}{\epsilon}\big)$ oracles calls to achieve the same. Similar accelerations can also be obtained for zeroth-order schemes [184], as briefly mentioned in Section 2.2.2.

Now we come to the heart of the technique. The following very nice theoretical argument aims at shedding some light on the acceleration mechanism; it was brought to our attention by Hardt [96] and we would like to repeat it here. Similar explanations can be found at other places

in the literature, see e.g. [59, 67, 215, 226].

Consider the quadratic function $f_5(\mathbf{x}) := \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{b}^T\mathbf{x}$, for a symmetric positive definite matrix; with gradient $\nabla f(\mathbf{x}) = A\mathbf{x}-\mathbf{b}$, optimum $\mathbf{x}^* = A^{-1}\mathbf{b}$ and condition number κ. For starting point $\mathbf{x}_0 = \mathbf{b}$, standard Gradient Descent calculates a sequence $(\mathbf{x}_k)_{k\geq 0}$ of iterates such that

$$\mathbf{x}_k = \sum_{i=0}^{k} \sigma(I_n - \sigma A)^i \mathbf{b}\,,$$

where $\sigma > 0$ is a constant step size, chosen such that the spectral radius[8] $\|I_n - \sigma A\| < 1$. Recall that $\frac{1}{x} = \sum_{i=0}^{\infty}(1-x)^i$ for $|x| < 1$. The analogous result also holds for matrices, $X^{-1} = \sum_{i=0}^{\infty}(I_n - X)^i$ if $\|I_n - X\| < 1$. Hence, Gradient Descent computes a degree k polynomial approximation to the inverse A^{-1}. The error term is of the order $O\big(\|(I_n - \sigma A)^k\|\big) = O\big((1-\kappa)^k\big)$, which is exactly the convergence rate of Gradient Descent.

Finding a better approximation to the function $(X)^{-1}$ will provide a faster algorithm. We are looking for a degree k polynomial q_k such that the residual error $r_k = \big\|A(A^{-1} - q_k(A))\big\| = \|I_n - Aq_k(A)\|$ is minimized. Or written differently: a polynomial of the form $p_k(X) = I_n - Xq_k(X)$ with $p_k(0_n) = I_n$. Here 0_n denotes the n-dimensional all-zero matrix. The value $p_k(A)$ depends only on the eigenvalues of A (see e.g. [104]), therefore p_k should map the eigenvalues of A as close to zero as possible. Approximation by the Chebyshev polynomial[9] T_k of degree k yields the residual error $r_k = O\big((1-\kappa^{1/2})^k\big)$, a quadratic improvement in the convergence rate. The Chebyshev polynomial T_k of degree k can recursively be computed from the two lower degree polynomials T_{k-1} and T_{k-2} by the recurrence $T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$ for $k > 1$. Consequently, an algorithm implementing this idea does not need to access the gradients of all previous iterates $(\mathbf{x}_i)_{i=0}^{k}$, but the two belonging to $\mathbf{x}_k$ and $\mathbf{x}_{k-1}$ are enough.

This convergence rate cannot further be improved. Nesterov [182] gives an example of a quadratic function for which this convergence rate cannot be beaten. The eigenvalues of this function are—when restricted to finite dimension—the roots of U_n, the Chebyshev polynomials of the

[8] Though we use standard notation in this paragraph, not everything has been properly introduced up to now. We apologize to the non-expert readers and direct them to Section 2.5 below.

[9] Defined as $T_0(x) = 0$, $T_1(x) = x$, and $T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$ for $k > 1$.

second kind.[10] The idea of acceleration by Chebyshev's method is also present in other fields. For instance in Lanczos method [47, 137, 139] to compute extremal eigenvalues or Chebyshev's iterative method to solve linear systems [57, 81, 102].

2.3.4 Constraints and Non-Smooth Functions

The optimization problem ($\mathrm{OPT_R}$) is a constrained optimization problem. Here the constraints describe the very simple set $B_0(R)$, but in general it could be an arbitrary convex set Q. A simple approach is the following: use any standard algorithm that works for the unconstrained problem (OPT), and if an iterate happens to fall outside of the feasible domain Q, project it back. However, this only works if Q is simple enough to allow for efficient computation of the projection. In general, projection onto a convex set is itself a constrained optimization problem (see e.g. [181])—and that is what we aim to solve.

There are two aspects to constrained black-box optimization: (i) if the set Q is *known*, thus not given as a black-box, we can reduce the constrained optimization problem in certain cases to an unconstrained one. We sketch these techniques briefly below. If (ii) the set Q is "hidden" in the objective function, for instance given as the set of points with finite function values, the situation is quite hopeless in the following sense. The geometry of Q has a major impact on the performance of any algorithm. A scheme can get stuck in a corner of the feasible set Q. Consider the positive orthant and let the origin be the current iterate. Simply to find a feasible point (and thus identifying the right orthant) takes exponential time. Lovász and Vempala [151] show that the Hit-and-Run algorithm can "escape" from a corner if the first iterate does not lie directly on the boundary, but sufficiently far away in the interior of Q. The same escape problem also arises on non-smooth functions: their level sets can have corners, and algorithmic schemes might get stuck there for exponential time. For this reason, we neither treat black-box constrained optimization problems nor non-smooth optimization problems in this thesis. Some aspects of derivative-free optimization with constrains are discussed e.g. in [8, 222].

The following methods (see e.g. [181, 187, 190]) can be applied if the set Q of constraints is known and sufficiently simple. The Penalty method solves a sequence of unconstrained optimization problems that are formulated such as to punish search points outside of the feasible

[10] $U_0(x) = 1$, $U_1(x) = 2x$, $U_{k+1} = 2xU_k(x) - U_{k-1}(x)$. $\frac{\mathrm{d}}{\mathrm{d}x}T_k(x) = (k+1)U_k(x)$.

domain Q by an increasing penalty term, forcing the optimal solution of the unconstrained problems to converge to the optimal solution of the constrained problem. This method requires that the function values of points outside the feasible set Q are well-defined. Barrier methods instead, force the iterates to stay inside the feasible set Q by punishing points close to the boundary (similar to the penalty methods). For instance if the feasible set describes a polytope, such barrier functions can be constructed by means of Dikin ellipsoids [181, pg. 182].

2.3.5 Randomization as a Design Principle

Our interest lies in the study of Random Pursuit algorithms of the form (2.2), $\mathbf{x}_{k+1} = \mathbf{x}_k + \sigma_k \mathbf{u}_k$, where typically the search directions are sampled from a probability distribution, and in some cases even the step sizes are influenced by randomness. One might wonder if this randomness is indeed necessary, or if there are equally good deterministic schemes. We don't know the answer to this. One of the reasons is certainly that the interesting problem classes of non-convex problems are not well-defined, and thus also this question.

One drawback of deterministic methods is that they often fail on very simple functions. For instance the Gradient Method gets immediately stuck in local minima, the Nelder-Mead simplex method may not even converge [258]. Practitioners observe that it is generally a good strategy to add some "random wiggles" to jump out of these minima or degeneracies. On one hand, randomization makes it harder to find examples where the methods certainly fail. This can also be observed with problems from other fields, like for instance the pivoting in the Simplex Method for Linear Programming. The Random Facet method performs well on most problems, but its worst-case behavior is (sub-)exponential [63, 66, 123, 124, 158]. On the other hand, there is some theoretical justification that this randomization indeed simplifies the complexity of non-convex optimization problems.

Let $\sigma > 0$ and $f\colon \mathbb{R}^n \to \mathbb{R}$ be a continuous function with bounded support. For $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, I_n)$ the *Gaussian smoothing* is defined as

$$f_\sigma(\mathbf{x}) := \mathbb{E}\left[f(\mathbf{x} + \sigma\mathbf{u})\right] , \quad \text{with} \quad \nabla f_\sigma(\mathbf{x}) = \frac{1}{\sigma}\mathbb{E}\left[f(\mathbf{x} + \sigma\mathbf{u})\mathbf{u}\right] . \quad (2.5)$$

This operation is also known under different names, like convolution with a Gaussian kernel or lowpass frequency filtering. In general, f_σ has nicer properties than f (see e.g. [53, 178, 184]). For instance if

f is Lipschitz continuous, then f_σ is differentiable with Lipschitz continuous gradient [184]. For the derivation of $\nabla f_\sigma(\mathbf{x})$ see [53, 184]. In general, f_σ is not only nicer than f in terms of smoothness, but also it is "more" convex. For instance Loog et al. [147] or Mohabi and Ma [163] investigated in which cases f_σ is a convex function even if f was not. The latter authors call a function *asymptotically convex* if $\lim_{\sigma\to\infty} f_\sigma$ is convex. For instance all functions with bounded support, that is problems $(\mathrm{OPT_R})$, or rapidly decaying functions are asymptotically convex. Therefore if an algorithm queries function values at points $\mathbf{x}_{k+1} = \mathbf{x}_k + \sigma\mathbf{u}_k$ for $\mathbf{u}_k \sim \mathcal{N}(\mathbf{0}, I_n)$, it "observes" function values and gradients of f_σ instead of f. If f_σ is convex, then the reasoning and optimization strategies for convex functions can be applied. As a technical difficulty, the parameter σ is typically changing over time, thus the algorithm observes different functions in every iteration (this is avoided in [184], as σ is set to a fixed value). This idea seems to be a promising way to analyze and justify the behavior of certain randomized optimization algorithms, but so far no concise analysis appeared in the literature. Intuitively, we might say that large values of σ will allow the algorithms to "jump" over local minima if the function has a "convex-like" global structure, but for small values of σ one cannot escape local minima any more, like in Gradient Descent.

2.4 Evolution Strategies

The term Evolution Strategies (ES) describes a certain class of bio-inspired iterative search schemes. Yet, the relation to biology is of a very abstract kind. In contrast to the Random Pursuit schemes that update only one search point in every iteration, the ES update *sets* of points. That is, the state in iteration k is described by a set of λ search points ("population"), where in general $\lambda \geq 1$. In every iteration, the objective function is evaluated at $\mu \geq 1$ new search points in the neighborhood of the old ones ("mutation"). To complete one iteration, the scheme picks a new set of λ points out of the total $(\mu + \lambda)$ points that are considered in the current iteration ("selection"). Such a strategy is called $(\mu + \lambda)$-ES. If the selection takes place only among the μ new search points, one writes (μ, λ)-ES instead. To formulate this process as a Random Pursuit algorithm we could for instance focus on the mean, or simply only the best search point in each iteration. For simplicity, we consider in the following only populations of size 1, i.e. (1+1)-ES.

For a comprehensive introduction to ES and some concise mutation and selection strategies see for instance [25, 27]. For instance in CMA-ES, the new search points are sampled from a normal distribution whose mean is the search point from the last iteration. An important feature of the selection mechanism in CMA-ES is the following: the function values are only used to compare the qualities of different search points and to identify the best ones (ranking). The concrete function values are not used anywhere else in the algorithm. This is in contrast to gradient-based schemes that might use finite-differences to determine step sizes, say. Whilst at first sight this might seem to be a handicap, the scheme becomes robust in the following sense: its behavior is exactly the same on the convex function $f(\mathbf{x}) = \|\mathbf{x}\|^2$ and on the non-convex function $g(\mathbf{x}) = \|\mathbf{x}\|^{1/2}$; in general on any strictly monotone transformation of f, i.e. $T(f(\mathbf{x}))$ for $T\colon \mathbb{R} \to \mathbb{R}$ strictly monotonically increasing. However, note that we typically study the convergence in the function value (2.1). This would not make sense for non-convex functions, like g, and one could for instance look at the distance of the current search point to the optimum instead.

Due to this invariance property, only the geometry of the level sets of the objective function has an impact on the algorithm's decision, but not the absolute function values. This is also the case for two other important features of CMA-ES that we will briefly sketch below. First, we discuss the step size adaptation mechanism and then we explain how CMA-ES adapts its sampling distribution to the objective function. For a more complete introduction to CMA-ES we suggest the tutorial [91].

2.4.1 Step Size Adaptation

In this section we discuss the idea behind the step size adaptation of CMA-ES or related (1+1)-ES. CMA-ES chooses the search direction from a multivariate normal distribution, i.e. $\mathbf{u}_k \sim \mathcal{N}(\mathbf{0}, I_n)$. In Section 2.3.1 we presented a step size adaptation scheme that is based on the success probability (2.3), i.e. the step size σ_k is selected such that the probability that $\mathbf{x}_k + \sigma_k \mathbf{u}_k$ has a lower function value than $\mathbf{x}_k$ is equal to a constant value c, $0 < c < \frac{1}{2}$. We have already observed that the success probability increases if σ_k is decreased. This suggests the following scheme to determine a step size σ_k that *approximately* satisfies condition (2.3): If the empirically observed success probability is smaller than c, increase the step size by a constant factor; otherwise decrease it. A straightforward implementation of this idea is the following

scheme:

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_k + \sigma_k \mathbf{u}_k & \text{if } f(\mathbf{x}_k + \sigma_k \mathbf{u}_k) \leq f(\mathbf{x}_k), \\ \mathbf{x}_k & \text{otherwise,} \end{cases} \tag{2.6}$$

and

$$\sigma_{k+1} = \begin{cases} \sigma_k \cdot a & \text{if } f(\mathbf{x}_k + \sigma_k \mathbf{u}_k) \leq f(\mathbf{x}_k), \\ \sigma_k \cdot b & \text{otherwise,} \end{cases} \tag{2.7}$$

where $a > 1$ and $b < 1$ are some parameters. For instance $a = e^{1/3}$ and $b = e^{-c/(3(1-c))}$ were picked in our numerical study [235]. From a theoretical point of view, the exact value c in scheme (2.3) is not important for convergence on quadratic functions [111–113], but it has an impact on the convergence rate.[11] If the success probability c is picked close to $1/2$ the steps tend to be too small, and on the other hand if c is too small, then there is no progress in most of the iterations; thus an optimal trade-off has to be found. Adaptive step size control was first presented by Schumer and Steiglitz [220]. They suggest a value of $c = 0.27$, based on theoretical investigations on the function $f_2(\mathbf{x}) = x_1^2 + x_2^2$. Rechernberg [209] suggests a value of $c = 1/5$ and his scheme is nowadays referred to as "1/5-th success rule". The scheme (2.7) is very aggressive: the step size is changed upon every success or failure. For the original variants [209, 220, 221] the authors suggested to evaluate the empirical success rate over the last (at least) n iterations. This regularization is also present in CMA-ES through means of the *Evolution Path* $\mathbf{p}_k$ [91, 93, 95]. This variable accumulates the steps $\sigma_k \mathbf{u}_k$ over the past iterations and is updated according to $\mathbf{p}_{k+1} = d \cdot \mathbf{p}_k + e \cdot \sigma_k \mathbf{u}_k$ for parameters d, e. Specific values can be found in [91]. Typically, $d \approx \left(1 - \frac{1}{n}\right)$, hence the contribution of a single step $\sigma_k \mathbf{u}_k$ is reduced by a constant factor every n iterations. Based on the length of $\mathbf{p}_k$, the step size is either decreased or increased, for instance similar to (2.7).

The right panel of Figure 2.1 on page 26 depicts a (1+1)-ES with the adaptive step size adaptation (2.7) described above. In our description above, the search directions where sampled in every iteration from the same distribution, but naturally, the variable metric approach is very common in Evolution Strategies. Slightly different approaches are used for instance in Gaussian Adaptation [167] or CMA-ES. We introduce the latter mechanism in the next section below.

[11] Note that so far there are no theoretical convergence results, yet rates, for the specific implementation (2.7) of the scheme (2.3).

2.4.2 Covariance Estimation

The most important feature of CMA-ES is the fact that it does not necessarily sample the search directions from the isotropic distribution $\mathcal{N}(\mathbf{0}, I_n)$, but from a multinormal distribution $\mathcal{N}(\mathbf{0}, \Sigma)$ with positive definite covariance Σ. The covariance Σ is updated in every iteration, to favour better search directions. This makes CMA-ES a zeroth-order variable metric scheme.

The covariance Σ_k in iteration k is updated by a rank-one update: $\Sigma_{k+1} = (1 - \tau) \cdot \Sigma_k + \tau \cdot \mathbf{y}_k \mathbf{y}_k^T$, where τ is a damping parameter, typically roughly n^{-2} [95], and $\mathbf{y}_k$ is a successful search direction. For instance $\mathbf{y}_k = \sigma_k \mathbf{u}_k$, for a successful search step $\sigma_k \mathbf{u}_k$ [95, 133]. Other variants use the evolution path $\mathbf{y}_k = \mathbf{p}_k$ instead [135], or a combination of both [91, 95]. Especially, if the population size is larger than one, a weighted combination of all μ (successful) steps can be used to update Σ_k in a similar manner, that is a rank-μ update instead of only rank-one [93]. Recently, Akimoto et al. [5] showed that the covariance matrix update of CMA-ES can be interpreted as a Monte Carlo approximation to an underlying natural gradient [26, 252].

2.5 Notation and Definitions

Here we introduce some basic notation and definitions, and review some well known facts from linear algebra, see e.g. [80, 107].

2.5.1 Vector Spaces, Norms and Eigenvalues

We denote by $\mathbb{R}^n$ the standard n-dimensional vector space over the real numbers $\mathbb{R}$, with inner product $\langle \mathbf{x}, \mathbf{y} \rangle := \mathbf{x}^T \mathbf{y}$ for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and induced *Euclidean norm* $\|\mathbf{x}\|_2 = \langle \mathbf{x}, \mathbf{x} \rangle^{1/2}$. If there is no risk of confusion, we omit the index and write $\|\mathbf{x}\| := \|\mathbf{x}\|_2$.

We denote by $\mathbb{R}^{m \times n}$ the set of all real $m \times n$ matrices. An inner product is defined by $\langle A, B \rangle = \mathrm{Tr}[A^T B] = \mathrm{Tr}[B A^T]$ for $A, B \in \mathbb{R}^{m \times n}$ and the *Frobenius norm* by $\|A\|_F = \langle A, A \rangle^{1/2}$. The set of symmetric matrices $A \in \mathbb{R}^{n \times n}$ with $A = A^T$ is denoted by SYM_n. A symmetric matrix $A \in \mathrm{SYM}_n$ is positive definite if $\mathbf{x}^T A \mathbf{x} > 0$ for all $\mathbf{x} \in \mathbb{R}^n \setminus \{0\}$ and we write $A \in \mathrm{PD}_n$. Every matrix $A \in \mathrm{PD}_n$ has a positive root, that is, there exists a matrix $A^{1/2} \in \mathrm{PD}_n$ with $A^{1/2} A^{1/2} = A$. The set PD_n is a convex cone, and a partial ordering is defined by the Löwner ordering $A \preceq B$ if $B - A \in \mathrm{PD}_n$.

For a symmetric matrix $A \in \mathrm{SYM}_n$ we denote by $\lambda(A) = \{\lambda_i(A) \in \mathbb{R}, i = 1, \ldots, n\}$, the set of eigenvalues of A, that is the set of numbers satisfying $\lambda_i(A)\mathbf{v}_i = A\mathbf{v}_i$, for pairwise orthogonal eigenvectors $\|\mathbf{v}_i\| = 1$, $i = 1, \ldots, n$. By the extremal characterization of eigenvalues, we have $\lambda_{\min}(A) = \min\{\mathbf{x}^T A\mathbf{x} : \|x\| = 1\}$ and $\lambda_{\max}(A) = \max\{\mathbf{x}^T A\mathbf{x} : \|x\| = 1\}$ for the smallest and largest eigenvalue of A. The matrices A and A^{-1} have the same set of eigenvectors, their eigenvalues are pairwise reciprocal, especially, $\lambda_{\max}(A) = \lambda_{\min}^{-1}(A^{-1})$. For any two matrices $A, B \in \mathbb{R}^{m\times n}$, the sets of the nonzero eigenvalues of the products AB^T and $B^T A$ are equal, see e.g. [204, Prop. 13.2].

The *spectral norm* of a matrix $A \in \mathbb{R}^{m\times n}$ is the operator norm induced by the Euclidean norm, that is,

$$\|A\|_2 = \sup_{\mathbf{x}\in\mathbb{R}^n, \|\mathbf{x}\|=1} \|\mathbf{x}\| = \lambda_{\max}(A^T A)^{1/2}\,.$$

The spectral norm and the Frobenius norm are topologically equivalent, that is $\|A\|_2 \le \|A\|_F \le \sqrt{r}\,\|A\|_2$, where $r \le \min\{m, n\}$ denotes the rank of A.

2.5.2 Condition Number

The *condition* number of a symmetric matrix $A \in \mathrm{SYM}_n$ is defined as $\kappa(A) := \|A\|_2 \left\|A^{-1}\right\|_2 = \kappa(A^{-1})$ and for $B \in \mathrm{PD}_n$ we have $\kappa(B) = \lambda_{\max}(B)\lambda_{\min}^{-1}(B)$. The *relative condition number* is defined as $\kappa_{\mathrm{F}}(A) := \left\|A^{-1}\right\|_2 \cdot \|A\|_F$. The condition number $\kappa(A)$ only depends on the two extremal eigenvalues, but not on the full eigenvalue spectra. For the algorithms considered in this thesis, this crude measure is often not sensitive enough to precisely describe the convergence behavior. As one alternative we suggest to measure the conditioning my means of the *average* eigenvalue instead of only the maximal one, that is the quantity $\frac{1}{n}\mathrm{Tr}[A]\lambda_{\min}^{-1}(A)$. For technical reasons (see Example 4.12 on page 65) we define the quantity $\kappa_{\mathrm{T}}(A) := \left(\mathrm{Tr}[A]\lambda_{\min}^{-1}(A) + 2\right)(n+2)^{-1}$.

Lemma 2.1. *For $A \in \mathrm{PD}_n$, we have*

$$\kappa_{\mathrm{T}}(A) \le \frac{1}{n}\mathrm{Tr}[A]\lambda_{\min}^{-1}(A) \le \kappa(A)\,.$$

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

2.5.3 Quadratic Norms

For a square matrix $A \in \mathbb{R}^{n\times n}$ we can define a quadratic form $Q_A(\mathbf{x}) = \mathbf{x}^T A\mathbf{x}$. Note that $\mathbf{x}^T A\mathbf{x} = \mathbf{x}^T\big(\frac{A+A^T}{2}\big)\mathbf{x}$, since the contribution of the asymmetric part $\big(\frac{A-A^T}{2}\big)$ vanishes by symmetry. We may thus assume without loss of generality $A \in \mathrm{SYM}_n$. If $A \in \mathrm{PD}_n$, the quadratic form Q_A defines a norm on $\mathbb{R}^n$, and write $\|\mathbf{x}\|_A = Q_A(\mathbf{x})^{1/2}$. We denote the unit sphere induced by this norm by $S_A^{n-1} := \{\mathbf{x} \in \mathbb{R}^n \colon \|\mathbf{x}\|_A = 1\}$, i.e. for $A = I_n$, the n-dimensional identity matrix, the sphere $S_{I_n}^{n-1} = S^{n-1}$ is just the standard unit sphere. The norm $\|\cdot\|_A$ induces a metric on $\mathbb{R}^n$, and we refer to this metric as the metric induced by A, or simply the metric A. In statistics this metric is also known as the Mahalanobis metric. The norm $\|\cdot\|_A$ is topologically equivalent to the Euclidean norm $\|\cdot\| = \|\cdot\|_{I_n}$:

$$\lambda_{\min}(A)\,\|\mathbf{x}\|^2 \le \|\mathbf{x}\|_A^2 \le \lambda_{\max}(A)\,\|\mathbf{x}\|^2 \,. \tag{2.8}$$

This follows directly from the extremal characterization of the eigenvalues, see Section 2.5.1 above. Note that both inequalities are tight, equality holds for $\mathbf{x} = \mathbf{v}_{\min}$ and $\mathbf{x} = \mathbf{v}_{\max}$, the eigenvectors corresponding the minimal and maximal eigenvalue of A, respectively. A generalization of this inequality is given in the following lemma.

Lemma 2.2. *Let $A \in \mathrm{SYM}_n$, $B \in \mathrm{PD}_n$, and $\mathbf{x} \in \mathbb{R}^n$ with $\mathbf{x} \neq \mathbf{0}$. Then*

$$\lambda_{\min}(B^{-1}A) \le \frac{\|\mathbf{x}\|_A^2}{\|\mathbf{x}\|_B^2} \le \lambda_{\max}(B^{-1}A)\,,$$

and both inequalities are tight.

Proof. The claim follows by reduction to (2.8). A proof can be found in [204, Prop. 18.3]. As we will use this statement frequently, the proof can also be found in the appendix on page 115. □

2.5.4 Function Classes and Quadratic Bounds

We now introduce some important inequalities that are useful for the subsequent presentation. A function $f\colon \mathbb{R}^n \to \mathbb{R}$ is in C^k if the first k derivatives all exist and are continuous. Mostly, we will consider convex functions $f \in C^1$. Smooth *convex* functions satisfy

$$f(\mathbf{y}) \ge f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle\,, \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n\,. \tag{2.9}$$

In smooth convex optimization one often assumes some additional regularity of the objective function, for instance that the curvature of $f \in C^1$ is bounded (cf. [34, 178, 182]). By this we mean that for some constant L (and some fixed metric $A \in \mathrm{PD}_n$),

$$|f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle| \leq \frac{L}{2} \|\mathbf{x} - \mathbf{y}\|_A^2 , \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n . \quad (2.10)$$

We will also refer to this inequality as the *quadratic upper bound*. It means that the deviation of f from any of its linear approximations can be bounded by a quadratic function. We denote by $C_L^1(A)$ the class of (at least once) differentiable convex functions for which the quadratic upper bound holds with parameter L (with respect to metric A). For smooth convex functions, the absolute value in condition (2.10) can be omitted due to (2.9). In the standard literature (cf. e.g. [34, 178, 182]) the curvature (2.10) is typically defined with respect to $A = I_n$. Here, we allow for a fully quadratic model given by the matrix $A \in \mathrm{PD}_n$. The class $C_L^1(A)$ comprises two important classes of functions: (i) smooth convex functions with L-Lipschitz continuous gradients (see e.g. [184, Lem. 1.2.3]) and by Taylor expansion we find (ii) twice differentiable convex functions with Hessian matrix $\nabla^2 f(\mathbf{x}) \preceq L \cdot A$ for all $\mathbf{x} \in \mathbb{R}^n$.

A differentiable function is *strongly convex* with positive parameter m if the *quadratic lower bound*

$$f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle \geq \frac{m}{2} \|\mathbf{y} - \mathbf{x}\|_A^2 , \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^n , \quad (2.11)$$

holds. We write $f \in C_{m,L}^1(A)$ with respect to metric $A \in \mathrm{PD}_n$ if $f \in C_L^1(A)$ satisfies (2.11) with parameter m. The ratio $\frac{L}{m}$, the *condition number* of f, measures the deviation of f from a quadratic function. If $L = m$, then f is quadratic. An important class of strongly convex functions are twice differentiable convex functions with Hessian matrix $\nabla^2 f(\mathbf{x}) \succeq m \cdot A$ for all $\mathbf{x} \in \mathbb{R}^n$.

Let $\mathbf{x}^*$ be the unique minimizer of a strongly convex function f with parameter m. Then equation (2.11) implies this useful relation:

$$\frac{m}{2} \|\mathbf{x} - \mathbf{x}^*\|_A^2 \leq f(\mathbf{x}) - f(\mathbf{x}^*) \leq \frac{1}{2m} \|\nabla f(\mathbf{x})\|_{A^{-1}}^2 , \quad \forall \mathbf{x} \in \mathbb{R}^n . \quad (2.12)$$

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

from (2.11) via

$$\begin{aligned} f(\mathbf{x}^*) &\geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{x}^* - \mathbf{x} \rangle + \frac{m}{2} \|\mathbf{x}^* - \mathbf{x}\|_A^2 \\ &\geq f(\mathbf{x}) + \min_{\mathbf{y} \in \mathbb{R}^n} \left(\langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{m}{2} \|\mathbf{y} - \mathbf{x}\|_A^2 \right) \\ &= f(\mathbf{x}) - \frac{1}{2m} \|\nabla f(\mathbf{x})\|_{A^{-1}}^2 \end{aligned}$$

by standard calculus.

2.5.5 Probability Distributions

The *multivariate normal* distribution arises from independent and identically distributed (i.i.d.) standard normals. The vector $\mathbf{u} \in \mathbb{R}^n$ is multivariate normally distributed with mean $\boldsymbol{\mu} \in \mathbb{R}^n$ and covariance $\Sigma = CC^T \in \mathrm{PD}_n$, i.e., $\mathbf{v} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ if $\mathbf{u} = \boldsymbol{\mu} + C\mathbf{v}$ for $\mathbf{v} \in \mathbb{R}^n$ with $v_i \sim \mathcal{N}(0,1)$ for $i = 1, \ldots, n$.

For any compact subset $C \subset \mathbb{R}^n$, we can define the uniform distribution over C. We will slightly abuse the notation and write $\mathbf{u} \sim C$ for a uniformly distributed vector from C. For a discrete set D, we write $\mathbf{u} \sim_w D$ for weighted distributions on D, that is, $\mathbf{u} = \mathbf{d} \in D$ with probability proportional to $w(\mathbf{d})$, for a positive function $w \colon D \to [0, \infty)$. For general sets, weighted distributions can be defined by means of density functions.

Definition 2.3 (Spherical Distribution)**.** *The vector $\mathbf{u} \in S^{n-1}$ is uniformly distributed on the unit sphere S^{n-1}, i.e. $\mathbf{u} \sim S^{n-1}$ if $\mathbf{u} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$ for $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, I_n)$.*

We now define an important class of Elliptical distributions, i.e. probability distributions on the unit sphere S_A^{n-1} for different metrics $A \in \mathrm{PD}_n$. In general, these distributions are different from the uniform distribution on S_A^{n-1}.

Definition 2.4 (Elliptical Distribution)**.** *The vector $\mathbf{u} \in S_A^{n-1}$ is elliptically distributed on the unit sphere S_A^{n-1} in metric $A \in \mathrm{PD}_n$ if $\mathbf{u} = C\mathbf{v}$ for $\mathbf{v} \sim S^{n-1}$ and $CC^T = A^{-1}$. We write $\mathbf{u} \sim S_A^{n-1}$.*

Remark 2.5. *Let $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, A^{-1})$ for $A \in \mathrm{PD}_n$. The random vector $\mathbf{v} = \frac{\mathbf{u}}{\|\mathbf{u}\|_A}$ is elliptically distributed on S_A^{n-1}.*

Proof. Let $A^{1/2}$ denote the symmetric positive definite root of A. Then $\mathbf{u} = A^{-1/2}\mathbf{w}$ for $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, I_n)$. Therefore $\frac{A^{1/2}\mathbf{u}}{\|\mathbf{u}\|_A} = A^{1/2}\mathbf{v}$ is spherically distributed, and by Definition 2.4 the vector $A^{-1/2}(A^{1/2}\mathbf{v}) = \mathbf{v}$ is elliptically distributed on S_A^{n-1}. □

2.6 Benchmark Functions

Table 2.3 below lists all functions that will be used in this thesis to benchmark the presented algorithms. We also list condition number and trace of the Hessians of the quadratic functions.

description	$\kappa(f)$	$\mathrm{Tr}[f]$
$f_{\text{two}} = \frac{1}{2}\left(\sum_{i=1}^{\lceil \frac{n}{2} \rceil} x_i^2 + L\sum_{i=\lceil \frac{n}{2} \rceil+1}^{n} x_i^2\right)$	L	$\frac{n(L+1)}{2}$
$f_{\text{flat}} = \frac{1}{2}\left(x_1^2 + \frac{L}{2}\sum_{i=2}^{n-1} x_i^2 + Lx_n\right)$	L	$\frac{n(L+1)}{2}$
$f_{\text{exp}} = \frac{1}{2}\left(\sum_{i=1}^{n} L^{\frac{i-1}{n-1}} x_i^2\right)$	L	$\approx \frac{n(L-1)}{\ln L}$
$f_{\text{lin}} = \frac{1}{2}\left(\sum_{i=1}^{n}\left(1 + (i-1)\frac{(L-1)}{(n-1)}\right) x_i^2\right)$	L	$\frac{n(L+1)}{2}$
$f_{\text{rosen}} = \sum_{i=1}^{n-1}\left(100 \cdot \left(x_i^2 - x_{i+1}\right)^2 + (x_i - 1)^2\right)$	-	-

Table 2.3: List of Benchmark functions for curvature parameter $L \geq 1$. All functions are quadratic, except the non-convex f_{rosen}.

It's nonconvergent!

Prof. Farnsworth
Futurama Episode 105: Benderama
M. Groening and D. X. Cohen

Chapter 3

Convergence

In this chapter we study convergence and especially *convergence rates* of search algorithms on smooth convex functions. For this, we study the *sequence* $(\mathbf{x}_k)_{k\geq 0}$ of search points that is generated by an algorithm. By looking only at the sequence $(\mathbf{x}_k)_{k\geq 0}$ of search points, we don't have to specify which exact algorithm generated this sequence. Hence, most of the results hold for arbitrary search schemes (that satisfy the quality conditions that we formulate below). We discuss specific algorithmic schemes in Chapter 4.

Ideally, the sequence $(\mathbf{x}_k)_{k\geq 0}$ just corresponds to all points at which algorithms evaluate the function $f(\mathbf{x}_k)$. But this does not need to be the case. For instance, $\mathbf{x}_k$ could also denote the best search point that has been discovered so far, or some summary statistics to describe the state of population based algorithms. We see that in general, the index k does not need to coincide with the complexity as defined in Section 2.3, that is, the number of function evaluations, but it rather denotes the current *iteration* of the algorithm. Any iteration could comprise many function evaluations, for instance to solve algorithmic subtasks, like gradient estimation [184] or performing a line search. We would like to interpret the difference between two successive iterates as a *step* of the algorithm, although this does not need literally to be the case.

Definition 3.1 (Local Search Scheme (LSS))**.** *Let $(\sigma_k)_{k\geq 0}$ be a sequence of scalars $\sigma_k \in \mathbb{R}$ and let $(\mathbf{u}_k)_{k\geq 0}$ be a sequence of vectors $\mathbf{u}_k \in \mathbb{R}^n$. A* Local Search Scheme *starting at $\mathbf{x}_0 \in \mathbb{R}^n$ with step sizes $(\sigma_k)_{k\geq 0}$ and search directions $(\mathbf{u}_k)_{k\geq 0}$ is a sequence $(\mathbf{x}_k)_{k\geq 0}$ of vectors $\mathbf{x}_k \in \mathbb{R}^n$*

satisfying

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \sigma_k \mathbf{u}_k \,. \tag{3.1}$$

Note that for a sequence $(\mathbf{x}_k)_{k\geq 0}$, the step sizes and search directions are not uniquely defined.

We introduce a simple and lightweight framework to analyze Local Search Schemes. Our framework decomposes the search procedure into two parts: first (i) the quality of the search directions and (ii) the quality of the step size. As discussed in Section 2.3.2, we measure the quality of the search directions in terms of the squared angle between the search direction and the gradient vector at the current search point. We generalize condition (2.4) to arbitrary metrics that are defined by a positive definite matrix A, see Section 2.5.3. The quality of the step sizes is measured with a specific sufficient decrease condition that relates the one step progress to the squared norm of the gradient at the current search point.

In Sections 3.1 and 3.2 we introduce the sufficient decrease condition and show convergence of the Local Search Scheme. Our presentation is extending the one of Karmanov [126, 127, 264]. We revisit Karmanov's result in this chapter and we enhance his results in a number of ways. The first convergence result for the Local Search Scheme—Theorem 3.3 below—depends only on the sequence of search points $(\mathbf{x}_k)_{k\geq 0}$, but is entirely independent of the algorithm that generated this sequence. This result is of very general nature, and can be viewed as an *a posteriori* analysis of the sequence $(\mathbf{x}_k)_{k\geq 0}$. Despite some mild assumptions, we do not yet specify *how* the search directions are generated, and *how* the step sizes are computed. This will be the subject of Chapter 4.

One might be more interested in the *predicted* convergence behavior of a sequence $(\mathbf{x}_k)_{k\geq 0}$ that *will be* generated by a specific algorithm. Therefore, we study convergence in expectation for randomized schemes in Section 3.3. We will assume that the search directions are chosen independently at random from some (fixed) probability distribution, and that the step sizes are chosen to satisfy the sufficient decrease condition. This could for instance be achieved with a line search procedure.

Let us make a concrete example to illustrate these convergence results. One important instance of a Random Pursuit algorithm is the scheme that selects the search directions independently from the unit sphere $\mathbf{u}_k \sim S^{n-1}$ and determines the step size by an (exact) line search. Theorem 3.6 shows that this scheme needs $O\big(\frac{L}{m}\mathbb{E}[\beta^2]^{-1}\ln\frac{1}{\epsilon}\big)$ iterations to find an ϵ-approximate solution on a strongly convex func-

tion $f \in C^1_{m,L}(I_n)$. The parameter β^2 describes the quality of the search directions in terms of the squared angle condition (2.4) and depends on the sampling distribution. In Chapter 4 we will show $\mathbb{E}[\beta^2]^{-1} = n$, for the uniform distribution on S^{n-1}. Hence, the running time scales linearly in the dimension n, and only logarithmically in $\frac{1}{\epsilon}$. This dependence on the accuracy ϵ is also called *linear* convergence. The running time increases on functions that are not strongly convex. For instance for $f \in C^1_L(I_n)$, the running time is $O\big(\frac{nL}{\epsilon}\big)$ instead, as we will show in Theorem 3.7.

Strong convexity is not only a global property, but a local property as well: a function needs to have positive curvature everywhere to be strongly convex. Therefore, if a function is linear in the neighborhood of one point in the domain we cannot expect linear convergence in general. However, in Theorem 3.12 we will show that Random Pursuit algorithms with a line search procedure can pass these linear plateaus quickly, implying linear convergence on a slightly broader class than just the strongly convex functions.

The theorems do not only describe the convergence for algorithms with exact line search oracles, but also for inexact line search procedures. We distinguish relative and absolute errors. Whilst the first ones only slow down the convergence (but do not hamper it), the latter are more serious: on strongly convex functions they prevent convergence below some accuracy level $\epsilon' > 0$, and on general convex functions they can also lead to divergence. However, this cannot happen if the line search is implemented in such way that it never accepts worse search points, i.e. points with a higher function value than the current one. But the search might stall (at a certain accuracy level).

The discussion of randomized schemes is complemented by a few concentration inequalities. We discuss such bounds in Section 3.6.

3.1 Local Search with Sufficient Decrease

We now formally define a first sufficient decrease condition.

Definition 3.2 (Sufficient Decrease)**.** *Let $(\gamma_k)_{k\geq 0}$ be a sequence of nonnegative numbers* (gains) *with $\gamma_k \in \mathbb{R}$, let $(\epsilon_k)_{k\geq 0}$ be a sequence of nonnegative numbers* (errors)*. The Local Search Scheme $(\mathbf{x}_k)_{k\geq 0}$ with search directions $(\mathbf{u}_k)_{k\geq 0} \in \mathbb{R}^n$ and step sizes $(\sigma_k)_{k\geq 0}$ satisfies the sufficient decrease condition on the function $f \in C^1_L(A)$ for a given positive*

definite matrix A (metric) *and the tuple* $((\gamma_k)_{k\geq 0}, (\epsilon_k)_{k\geq 0})$ *if*

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) - \gamma_k \frac{\beta_k^2}{2L} \|\nabla f(\mathbf{x}_k)\|_{A^{-1}}^2 + \epsilon_k \,, \tag{D1}$$

for every $k \geq 0$. *Here* $\beta_k = \beta_A(\nabla f(\mathbf{x}_k), \mathbf{u}_k)$ *measures the quality of the search directions* $\mathbf{u}_k$ *by the generalized angle condition:*

$$\beta_k = \beta_A(\nabla f(\mathbf{x}_k), \mathbf{u}_k) := \frac{\langle \nabla f(\mathbf{x}_k), \mathbf{u}_k \rangle}{\|\nabla f(\mathbf{x}_k)\|_{A^{-1}} \|\mathbf{u}_k\|_A} \,. \tag{A1}$$

The sufficient decrease condition (D1) quantifies the quality of each step. The β_k parameters are uniquely defined by the search directions and the current position $\mathbf{x}_k$. The quality of the step sizes is expressed by the gains γ_k and the errors ϵ_k. We see from condition (D1) that the decrease in function value is largest if the gains γ_k are as large as possible and the errors ϵ_k as small as possible. Consider the function $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2$. This quadratic function is in $C_1^1(I_n)$. The squared angle measure (A1), that is, β_k^2, equals 1 if iterate $\mathbf{x}_{k+1}$ lies on the line defined by the gradient direction $\nabla(f(\mathbf{x}_k)) = -\mathbf{x}_k$ and the search point $\mathbf{x}_k$. Observe $\frac{1}{2}\|\nabla f(\mathbf{x}_k)\|_{I_n}^2 = f(\mathbf{x}_k)$, thus this example shows that in general we cannot expect (D1) to hold for gains $\gamma_k > 1$. Further we note that, given a pair $(\mathbf{x}_k, \mathbf{x}_{k+1})$ of iterates and function f, the tuple (γ_k, ϵ_k) is not uniquely defined. We see in Theorem 3.3 below that large errors ϵ_k are typically worse than low gains γ_k. Thus if the sequence of function values $f(\mathbf{x}_k)$ is monotonically decreasing, it is best to express the sufficient decrease (D1) only by the gains, and set $\epsilon_k = 0$. However, it might not always be possible to provide such strong bounds.

3.2 Smooth Convex Functions

Now we study the evolution of the decrease over a finite number of steps.

Theorem 3.3. *Let* $(\mathbf{x}_k)_{k\geq 0}$ *denote a Local Search Scheme* (3.1) *with search directions* $(\mathbf{u}_k)_{k\geq 0}$ *and step sizes* $(\sigma_k)_{k\geq 0}$ *that satisfy the sufficient decrease condition* (D1) *with parameters* $((\gamma_k)_{k\geq 0}, (\epsilon_k)_{k\geq 0})$ *on a smooth function* $f \in C_L^1(A)$ *for a positive definite matrix* A. *For* $k > 0$ *denote* $f_k := f(\mathbf{x}_k) - f^*$, *where* $f^* := \min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$, *and let* $S_N := \sum_{k=0}^{N-1} \gamma_k \beta_k^2$ *for* $N > 0$. *Furthermore let* $R := \sup_{\mathbf{x},\mathbf{y}\in\mathbb{R}^n} \left\{ \|\mathbf{x} - \mathbf{y}\|_A \mid \max\{f(\mathbf{x}), f(\mathbf{y})\} \leq f(\mathbf{x}_0) \right\}$ *the diameter of the* $f(\mathbf{x}_0)$ *level set.*

(i) If $f \in C^1_{m,L}(A)$ strongly convex, then

$$f_N \leq f_0 \cdot \prod_{i=0}^{N-1} \left(1 - \frac{\gamma_k \beta_k^2 m}{L}\right) + C_N \leq f_0 \cdot \exp\left[-\frac{m}{L} S_N\right] + C_N \,,$$

where $C_N := \epsilon_{N-1} + \sum_{i=2}^{N} \prod_{j=N+1-i}^{N-1} (1 - \frac{m\gamma_j\beta_j^2}{L}) \epsilon_{N-i}$.

(ii) If $f \in C^1_L(A)$ convex, $R < \infty$, $S_N > 0$ and the errors $\epsilon_k = 0$ for $k = 0, \ldots, N-1$, then

$$f_N \leq \frac{2LR^2}{S_N} \,.$$

(iii) If $f \in C^1_L(A)$ convex, $R < \infty$, the gains lower bounded, $\gamma_k \beta_k^2 \geq \delta > 0$, and errors upper bounded, $\epsilon_k \leq \epsilon$, for $k = 0, \ldots, N-1$, then

$$f_N \leq \frac{Q}{N} + (N-1)\epsilon \,,$$

where $Q = \max\left\{2LR^2/\delta, f_0\right\}$.

Part (ii) of Theorem 3.3 was already shown by Karmanov [126] for the case $A = I_n$. A nice presentation can be found in [264]. The proof of parts (ii) and (iii) can be found in the appendix on page 117. We now proceed to prove part (i). The proof shows nicely how the quadratic lower bound (2.11) can be used.

Proof of part (i). From the quadratic lower bound (2.12) it follows

$$\|\nabla f(\mathbf{x}_k)\|^2_{A^{-1}} \geq 2m f_k \,,$$

together with sufficient decrease (D1) this yields

$$f_k - f_{k+1} + \epsilon_k \geq \frac{\gamma_k \beta_k^2 m}{2L} \|\nabla f(\mathbf{x}_k)\|^2_{A^{-1}} \geq \frac{\gamma_k \beta_k^2 m}{L} f_k \,, \tag{3.2}$$

for $k = 0, \ldots, N-1$. Let $\tau_k := \gamma_k m \beta_k^2 / L$. We deduce

$$f_{k+1} \leq (1 - \tau_k) f_k + \epsilon_k \,,$$

for $k = 0, \ldots, N-1$ and thus

$$f_N \leq \prod_{i=0}^{N-1} (1 - \tau_k) f_0 + C_N \,.$$

By rearranging we deduce the first inequality in part (i) of Theorem 3.3. The second one follows from $(1-x) \leq e^{-x}$ for all $x \in \mathbb{R}$. □

Remark 3.4. *If in part (i) of Theorem 3.3 the gains are lower bounded, $\gamma_k \beta_k^2 \geq \delta > 0$, and error upper bounded, $\epsilon_k \leq \epsilon$, for $k = 0, \ldots, N-1$, then $C_N \leq \frac{L}{m\delta}\epsilon$.*

Proof. We estimate

$$C_N \leq \epsilon \sum_{i=1}^{N} \left(1 - \frac{m\delta}{L}\right)^{(N-i)} \leq \epsilon \frac{L}{m\delta}\,. \qquad \square$$

Remark 3.5. *The bound in part (iii) of Theorem 3.3 becomes meaningless as $N \to \infty$ if $\epsilon > 0$. Nevertheless, for $N_{\text{opt}} = \sqrt{Q/\epsilon}$ the estimate becomes*

$$f_{N_{\text{opt}}} \leq 2\sqrt{\epsilon Q}\,.$$

3.3 Convergence in Expectation

In the previous Section 3.2 we discussed the convergence of the Local Search Scheme (3.1) where the sequence of iterates $(\mathbf{x}_k)_{k\geq 0}$ was fixed. Often, the algorithmic schemes generate *random* sequences, that is, $\mathbf{x}_{k+1}$ is a random variable that can depend on the previous iterates $(\mathbf{x}_i)_{i=0}^{k}$. Of course, for any fixed sequence of copies, or realizations, of these random variables we can just apply Theorem 3.3. However, we are not only interested in the value f_N for one specific realization of the random variables, but like to compute its *expected value*—if possible.

We will think of the random sequence $(\mathbf{x}_k)_{k\geq 0}$ as generated by sequences of random search directions $(\mathbf{u}_k)_{k\geq 0}$ and random gains and errors. By this we mean, that each $\mathbf{u}_k$ is an independent copy of a random variable $\bar{\mathbf{u}}_k$ in $\mathbb{R}^n$ with some fixed distribution π_k (which could depend on $(\mathbf{x}_i)_{i=0}^{k}$). In the simplest case, all $\mathbf{u}_k$ just follow the same distribution, for instance the uniform distribution on the unit sphere. In the following we also make the (strong) assumption, that these random variables are such that each realization of $(\mathbf{u}_k, \gamma_k, \epsilon_k)$ satisfies the sufficient decrease condition (D1). We comment on this assumption at the very end of this section, and in Section 3.4 below we show that all assumptions are satisfied if the steps are for instance generated by a line search.

To estimate $\mathbb{E}[f_N]$, we would ideally just take the expectations on both sides of the inequalities of Theorem 3.3. The theorem provides an upper bound on $\mathbb{E}[f_N]$ if we can compute a bound on $E[e^{-S_N}]$ and $\mathbb{E}[1/S_N]$ for part (i) and (ii), respectively. However, we can compute these expectations only in rare cases, and it is in general much easier to deal only with $\mathbb{E}[S_N]$ instead. One approach to deal with this situation is to establish concentration bounds on the random variable S_N. If S_N is concentrated around its mean $E[S_N]$, then for a smooth transformation $T\colon \mathbb{R} \to \mathbb{R}$ we can expect $E[T(S_N)] \approx T(\mathbb{E}[S_N])$. We follow this approach in Section 3.6 and show that we can indeed expect good enough concentration results for typical applications, so we can restrict our attention to (lower) bounds on $\mathbb{E}[S_N]$. However, note that we could not use this approach for the situation in part (iii) of Theorem 3.3. There we need a uniform lower bound $\gamma_k\beta_k^2 \geq \delta$. We would need unrealistic assumptions on the concentration of $\gamma_k\beta_k^2$ to provide such a lower bound for every $\gamma_k\beta_k^2$ for $k = 0, \dots, N-1$. We show that is suffices to have a lower bound on the expectations $\mathbb{E}[\gamma_k\beta_k^2] \geq \delta$.

In the remainder of this section, we extend the statements of part (i) and part (iii) of Theorem 3.3 to the randomized setting.

Theorem 3.6. *Let $(\mathbf{x}_k)_{k\geq 0}$ denote a Local Search Scheme* (3.1) *with independent random search directions $(\mathbf{u}_k)_{k\geq 0}$ and step sizes $(\sigma_k)_{k\geq 0}$, that satisfies the sufficient decrease condition* (D1) *with random parameters $((\gamma_k)_{k\geq 0}, (\epsilon_k)_{k\geq 0})$ on $f \in C^1_{m,L}(A)$ for a positive definite matrix A. Assume ϵ_k is independent of $\gamma_k\beta_k^2$ and let β_k, f_N, C_N and S_N be defined as in Theorem 3.3. Then*

$$\mathbb{E}[f_N] \leq f_0 \cdot \prod_{i=0}^{N-1}\left(1 - \frac{\mathbb{E}[\gamma_k\beta_k^2]m}{L}\right) + \mathbb{E}[C_N] \leq f_0 \cdot \exp\left[-\frac{m}{L}\mathbb{E}[S_N]\right] + D_N,$$

where $D_N := \mathbb{E}[\epsilon_{N-1}] + \sum_{i=2}^{N}\prod_{j=N+1-i}^{N-1}\left(1 - \frac{m\mathbb{E}[\gamma_j\beta_j^2]}{L}\right)\mathbb{E}[\epsilon_{N-i}]$.

Proof. The theorem follows from the first inequality in part (i) of Theorem 3.3. We condition on $(\mathbf{x}_k)_{k=0}^{N-1}$ and take the conditional expectation on both sides. This yields:

$$\begin{aligned}\mathbb{E}\left[f_N \mid (\mathbf{x}_k)_{k=0}^{N-1}\right] &\leq f_0 \cdot \mathbb{E}\left[\prod_{i=0}^{N-1}\left(1 - \frac{\gamma_k\beta_k^2 m}{L}\right) + C_N \;\middle|\; (\mathbf{x}_k)_{k=0}^{N-1}\right] \\ &= f_0 \cdot \Phi \cdot \left(\prod_{i=0}^{N-2}\left(1 - \frac{\gamma_k\beta_k^2 m}{L}\right) + D_{N-1}\right) + \mathbb{E}[\epsilon_{N-1}]\,,\end{aligned}$$

with $\Phi := \mathbb{E}\big[1 - \gamma_{N-1}\beta_{N-1}^2 m/L\big]$. By taking the conditional expectations on $(\mathbf{x}_k)_{k=0}^{N-2}$, $(\mathbf{x}_k)_{k=0}^{N-3}$, $\dots$, $(\mathbf{x}_0)$, the tower property of conditional expectations yields $\mathbb{E}\big[\cdots\mathbb{E}\big[\mathbb{E}\big[f_N \mid (\mathbf{x}_k)_{k=0}^{N-1}\big] \mid (\mathbf{x}_k)_{k=0}^{N-2}\big]\cdots \mid \mathbf{x}_0\big] = \mathbb{E}[f_N]$, and the first inequality follows. The second follows as in the proof of Theorem 3.3 from $(1-x) \le e^{-x}$ for all $x \in \mathbb{R}$. □

Theorem 3.7. *Let $(\mathbf{x}_k)_{k\ge0}$ denote a Local Search Scheme* (3.1) *with independent random search directions $(\mathbf{u}_k)_{k\ge0}$ and step sizes $(\sigma_k)_{k\ge0}$, that satisfies the sufficient decrease condition* (D1) *with random parameters $((\gamma_k)_{k\ge0}, (\epsilon_k)_{k\ge0})$ on $f \in C_L^1(A)$ for a positive definite matrix A. Assume ϵ_k is independent of $\gamma_k\beta_k^2$ and let β_k, f_N and R be defined as in Theorem 3.3. If $R < \infty$, the expected gains lower bounded, $\mathbb{E}[\gamma_k\beta_k^2] \ge \delta > 0$, and expected errors upper bounded, $\mathbb{E}[\epsilon_k] \le \epsilon$, for $k = 0, \dots, N-1$, then*

$$\mathbb{E}[f_N] \le \frac{Q}{N} + (N-1)\epsilon\,, \tag{3.3}$$

where again $Q = \max\big\{2LR^2/\delta, f_0\big\}$.

Proof. First, we derive $\mathbb{E}\big[f_N \mid (\mathbf{x}_k)_{k=0}^{N-1}\big]$, conditioning on $(\mathbf{x}_k)_{k=0}^{N-1}$. Because $\mathbf{x}_{N-1}$ is fixed, we can express $\mathbb{E}\big[f_N \mid (\mathbf{x}_k)_{k=0}^{N-1}\big]$ in terms of f_{N-1}. The derivation of the one step progress is identical to the derivation of equation (B.1) on page 118. We have

$$\begin{aligned}\mathbb{E}\left[f_N \mid (\mathbf{x}_k)_{k=0}^{N-1}\right] &\le \mathbb{E}\left[f_{N-1} - \frac{\gamma_{N-1}\beta_{N-1}^2}{2LR^2} f_{N-1}^2 + \epsilon_{N-1} \;\middle|\; (\mathbf{x}_k)_{k=0}^{N-1}\right] \\ &\le f_{N-1} - 2\tau f_{N-1}^2 + \epsilon\,,\end{aligned}$$

where $\tau = \delta/(2LR^2)$. We reformulate this bound with the same technique as in the proof of Theorem 3.3 on page 118 to read

$$\mathbb{E}\left[f_N \mid (\mathbf{x}_k)_{k=0}^{N-1}\right] \le (1 - 2h_{N-1})f_{N-1} + \frac{h_{N-1}^2}{\tau} + \epsilon\,,$$

for all $h_{N-1} \in \mathbb{R}^n$. Now, formally, we recursively apply the conditional expectations, conditioning on $(\mathbf{x}_k)_{k=0}^{N-2}$, $(\mathbf{x}_k)_{k=0}^{N-3}$, $\dots$, $(\mathbf{x}_0)$, and perform the same reformulations in every step. We end up with a bound on $\mathbb{E}[f_N]$ that depends on the free parameters $h_0, \dots h_{N-1}$. By setting $h_k = 1/(k+1)$ for $k = 0, \dots, N-1$, we obtain a recurrence that is exactly of the form as treated in Lemma A.9 on page 116. □

In this section we assumed that each realization of the random variables $(\mathbf{u}_k, \gamma_k, \epsilon_k)$ satisfied the sufficient decrease condition (D1). However, for the situation in Theorem 3.6 (strongly convex functions), we could rely on a relaxed condition instead. It is easily checked (see e.g. equation (3.2)) that a bound on the expected one step progress is sufficient. In the setting of Definition 3.2 this means

$$\mathbb{E}\left[f(\mathbf{x}_{k+1}) \mid (\mathbf{x}_i)_{i=0}^k\right] \leq f(\mathbf{x}_k) - \alpha_k^2 \frac{1}{2L} \|\nabla f(\mathbf{x}_k)\|_{A^{-1}}^2 + \mathbb{E}[\epsilon_k]\,, \quad \text{(E1)}$$

for parameters $(\alpha_k^2)_{k\geq 0}$. Certainly, the stronger conditions of Theorem 3.6 do imply (E1) for $\alpha_k^2 = \mathbb{E}[\gamma_k \beta_k^2]$. In the proof of Theorem 3.7 we used the sufficient decrease condition (D1) slightly differently, and the proof cannot easily be adapted for (E1). However, we will now proceed by showing that the strong assumptions on $(\mathbf{u}_k, \gamma_k, \epsilon_k)$ might not be that unrealistic after all. They can for instance be satisfied if an algorithmic scheme first generates a search direction $\mathbf{u}_k$ at random, but then carefully—for instance with a line search—selects a step size such that (D1) is satisfied for fixed parameters γ and ϵ.

3.4 Line Search with Sufficient Decrease

In this section, we show that sufficient decrease (D1) can be obtained by solving a one-dimensional optimization problem, i.e. a line search.

Definition 3.8 (Exact line search oracle)**.** *For $\mathbf{x} \in \mathbb{R}^n$, a convex function $f \in C^1$, and a direction $\mathbf{u} \in \mathbb{R}^n$, a function $\mathtt{LS}^f \colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ with*

$$f\left(\mathbf{x} + \mathtt{LS}^f(\mathbf{x}, \mathbf{u})\mathbf{u}\right) = \min_h f(\mathbf{x} + h\mathbf{u}) \quad (3.4)$$

is called an exact line search oracle.

Note that the value of $\mathtt{LS}^f(\mathbf{x}, \mathbf{u})$ does not necessarily follow just from the definition. The condition (3.4) might be satisfied for an interval $[\omega_1, \omega_2]$ of values in $\mathbb{R}$, out of which $\mathtt{LS}^f(\mathbf{x}, \mathbf{u})$ picks an element. However, on strongly convex function this interval reduces to a point and the value of $\mathtt{LS}^f(\mathbf{x}, \mathbf{u})$ is uniquely defined by f, $\mathbf{x}$ and $\mathbf{u}$. The exact line search oracle (3.4) is a rather idealistic concept, as we cannot except to have access to such an oracle for most applications. We now define (the more practical) inexact line search oracle.

Definition 3.9 (Inexact line search oracle)**.** *For $\mathbf{x} \in \mathbb{R}^n$, a convex function $f \in C^1$, direction $\mathbf{u} \in \mathbb{R}^n$, and parameters $\gamma \geq 0$, $\epsilon \geq 0$, let $\tilde{f} := \min_h f(\mathbf{x} + h\mathbf{u})$. A function $\mathtt{LS}^f \colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is called an (γ, ϵ)-line search oracle if $\mathbf{x}_+ := \mathbf{x} + \mathtt{LS}^f(\mathbf{x}, \mathbf{u})\mathbf{u}$ satisfies the following condition with parameters (γ, ϵ):*

$$f(\mathbf{x}_+) = f\left(\mathbf{x} + \mathtt{LS}^f(\mathbf{x}, \mathbf{u})\mathbf{u}\right) \leq (1-\gamma)f(\mathbf{x}) + \gamma\tilde{f} + \epsilon\,. \tag{D2}$$

It is not hard to see that (D2) implies (D1) for functions with a quadratic upper bound (2.10).

Lemma 3.10 (D2) $\Rightarrow$ (D1)**.** *Let $\mathbf{x} \in \mathbb{R}^n$, function $f \in C^1_L(A)$ for metric $A \in \mathrm{PD}_n$, search direction $\mathbf{u} \in \mathbb{R}^n$, step size $\sigma \in \mathbb{R}$, and $\mathbf{x}_+ = \mathbf{x} + \sigma\mathbf{u}$ satisfying the sufficient decrease condition* (D2) *for parameters (γ, ϵ). Then $\mathbf{x}_+$ satisfies* (D1) *for parameters (γ, ϵ).*

Proof. We use the quadratic upper bound (2.10) to derive an upper bound on $\tilde{f}$.

$$\begin{aligned}\tilde{f} = \min_{h\in\mathbb{R}} f(\mathbf{x} + h\mathbf{u}) &\leq f(\mathbf{x}) + \min_{h\in\mathbb{R}} \left(h\,\langle \nabla f(\mathbf{x}), \mathbf{u}\rangle + h^2 \frac{L}{2} \|\mathbf{u}\|_A^2 \right) \\ &\leq f(\mathbf{x}) - \frac{\langle \nabla f(\mathbf{x}), \mathbf{u}\rangle^2}{2L\,\|\mathbf{u}\|_A^2} = f(\mathbf{x}) - \frac{\beta_A^2(\nabla f(\mathbf{x}), \mathbf{u})}{2L} \|\nabla f(\mathbf{x})\|_{A^{-1}}^2\,,\end{aligned}$$

by the choice $h = -\frac{\langle \nabla f(\mathbf{x}), \mathbf{u}\rangle}{2L\|\mathbf{u}\|_A^2}$. The lemma now immediately follows from the definition (D2). □

In Definition 3.9 we define an inexact line search oracle for a fixed search direction $\mathbf{u}$. If $\mathbf{u}$ is sampled from a probability distribution, then condition (D2) might hold for different parameters (γ, ϵ) for each realization of $\mathbf{u}$. For this reason, we consider in this case also the parameters (γ, ϵ) as random variables and compute their expected values. Similar to our discussion at the end of the previous Section 3.3 we could have formulated condition (D2) slightly differently for random search directions $\mathbf{u}$. Instead of enforcing (D2) for every single copy of the random variable $\mathbf{u}$, we could consider the condition

$$\mathbb{E}\left[f(\mathbf{x} + \mathtt{LS}^f(\mathbf{x}, \mathbf{u})\mathbf{u}) \mid \mathbf{x}\right] \leq (1-\gamma)f(\mathbf{x}) + \gamma\mathbb{E}[\tilde{f}] + \mathbb{E}[\epsilon]\,, \tag{E2}$$

instead, where the expectation is over the choice of $\mathbf{u}$ and γ is now a fixed parameter. That is, equation (D2) needs to hold for every copy of

γ, but only on average for ϵ. By reasoning similar to Lemma 3.10, we see that (E2) implies (E1) for $\alpha^2 = \gamma\mathbb{E}\left[\beta_A^2\right]$. Consequently, this relaxed condition is enough to prove convergence on strongly convex functions—as before with condition (E1). However, in the next Lemma 3.11 below, we assume the stronger condition (D2).

3.5 Improvements for Line Search Oracle

We have shown that the line search oracle as introduced in Section 3.4 is a means to achieve the sufficient decrease condition (D1). In Theorem 3.6 we proved convergence on strongly convex functions. In this section we show that this result can be extended to a more general class of functions, if we assume the stronger decrease condition (D2) instead of only (D1). For this, we provide first a different bound on the one step progress, i.e. the quantity

$$f_k - \mathbb{E}\left[f_{k+1} \mid (\mathbf{x}_i)_{i=0}^k\right] .$$

3.5.1 One Step Progress

Lemma 3.11 (One step progress)**.** *Let $(\mathbf{x}_k)_{k\geq 0}$ denote a Local Search Scheme* (3.1) *with independent random search directions $(\mathbf{u}_k)_{k\geq 0}$ and step sizes $(\sigma_k)_{k\geq 0}$, that satisfies the sufficient decrease condition* (D2) *with random parameters $((\gamma_k)_{k\geq 0}, (\epsilon_k)_{k\geq 0})$ on convex $f \in C_L^1(A)$ for a positive definite matrix A. Let f_N be defined as in Theorem 3.3. Then*

$$\begin{aligned}\mathbb{E}\left[f(\mathbf{x}_{k+1}) \mid (\mathbf{x}_i)_{k=0}^k\right] \leq f(\mathbf{x}_k) + h_k \left\langle \nabla f(\mathbf{x}_k), \mathbb{E}\left[\gamma_k \left\langle T_k(\mathbf{x}_k), \mathbf{u}_k\right\rangle \mathbf{u}_k\right]\right\rangle \\ + \frac{h_k^2 L}{2}\mathbb{E}\left[\gamma_k \left\|\left\langle T_k(\mathbf{x}_k), \mathbf{u}_k\right\rangle \mathbf{u}_k\right\|_A^2\right] + \mathbb{E}[\epsilon_k] ,\end{aligned}$$

for every $T_k \colon \mathbb{R}^n \to \mathbb{R}^n$, $h_k \in \mathbb{R}$ and every $k \geq 0$.

Proof. We proceed as in the proof of Lemma 3.10 and use the quadratic upper bound (2.10) to derive an upper bound on the one step progress. For fixed $\mathbf{x}_k, \epsilon_k, \gamma_k$ and $\mathbf{u}_k$ we get

$$\begin{aligned} f(\mathbf{x}_{k+1}) &\leq (1-\gamma_k) f(\mathbf{x}_k) + \gamma_k \tilde{f} + \epsilon_k \\ &= (1-\gamma_k) f(\mathbf{x}_k) + \gamma_k \cdot \min_t f(\mathbf{x} + t\mathbf{u}_k) + \epsilon_k \\ &\leq f(\mathbf{x}_k) + \gamma_k \cdot \min_t \left(t \left\langle \nabla f(\mathbf{x}_k), \mathbf{u}_k\right\rangle + t^2 \frac{L}{2} \left\|\mathbf{u}_k\right\|_A^2 \right) + \epsilon_k , \end{aligned}$$

We set $t = h_k \langle T_k(\mathbf{x}_k), \mathbf{u}_k \rangle$ and take the expectation (conditioned on $(\mathbf{x}_i)_{i=0}^k$) on both sides. □

3.5.2 Improved Results

It is not necessary that the function f is strongly convex everywhere for linear convergence to hold. Theorem 3.12 below shows that convergence (at about a quarter of the rate of the one in Theorem 3.6) can be obtained assuming only a weaker condition. Let us recall that strong convexity with parameter m implies that

$$f(\mathbf{x}) - f(\mathbf{x}^*) \geq \frac{m}{2} \|\mathbf{x} - \mathbf{x}^*\|_A^2 \,, \forall \mathbf{x} \in \mathbb{R}^n \,, \tag{3.5}$$

where $\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x})$. Instead of strong convexity (2.11), the weaker condition (3.5) is enough for linear convergence. For example a function that is linear in the neighborhood of one single point $\mathbf{y} \in \mathbb{R}^n$ is not strongly convex. However, condition (3.5) could still hold if $\mathbf{y} \neq \mathbf{x}^*$. The theorem below is an immediate consequence of Lemma 3.11. Generalizations to other distributions (like the ones we will discuss in Chapter 4) can analogously be obtained.

Theorem 3.12. *Let $(\mathbf{x}_k)_{k\geq 0}$ denote a Local Search Scheme* (3.1) *with independent random search directions $(\mathbf{u}_k)_{k\geq 0}$ and step sizes $(\sigma_k)_{k\geq 0}$, that satisfies the sufficient decrease condition* (D2) *with random parameters $((\gamma_k)_{k\geq 0}, (\epsilon_k)_{k\geq 0})$ on $f \in C_L^1(A)$ for a positive definite matrix A. Let f_N be defined as in Theorem 3.3. Let γ_k and ϵ_k be independent and let γ_k and $\mathbf{u}_k$ be independent. Assume f has a unique minimizer $\mathbf{x}^* \in \mathbb{R}^n$ satisfying* (3.5) *with $m > 0$. Let $T_k \colon \mathbb{R}^n \to \mathbb{R}$ with $\mathbb{E}[\langle T_k(\mathbf{x}_k), \mathbf{u}_k \rangle \mathbf{u}_k] = (\mathbf{x}_k - \mathbf{x}^*)$. Suppose $\mathbb{E}[\|\langle T_k(\mathbf{x}_k), \mathbf{u}_k \rangle \mathbf{u}_k\|_A^2] \leq \theta \|\mathbf{x}_k - \mathbf{x}^*\|_A^2$ for all $k \geq 0$. Then*

$$\mathbb{E}\left[f_N\right] \leq f_0 \cdot \prod_{k=0}^{N-1} \left(1 - \frac{\mathbb{E}[\gamma_k] m}{4L\theta}\right) + D_N \,, \tag{3.6}$$

and $D_N := \mathbb{E}[\epsilon_{N-1}] + \sum_{i=2}^{N} \prod_{j=N+1-i}^{N-1} \left(1 - \frac{m\mathbb{E}[\gamma_j]}{4L\theta}\right) \mathbb{E}[\epsilon_{N-i}]$.

In Example 3.13 below, we present functions $(T_k)_{k\geq 0}$ that satisfy the assumptions of Theorem 3.12.

Proof. We use Lemma 3.11 together with the assumptions on T_k. We get

$$\mathbb{E}\left[f(\mathbf{x}_{k+1}) \mid (\mathbf{x}_i)_{k=0}^k\right] \leq f(\mathbf{x}_k) + h_k\mathbb{E}[\gamma_k]\left\langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{x}^*\right\rangle + \frac{h_k^2\mathbb{E}[\gamma_k]L\theta}{2}\left\|\mathbf{x}_k - \mathbf{x}^*\right\|^2 + \mathbb{E}[\epsilon_k]\,,$$

for any parameters $h_k \in \mathbb{R}$. Using convexity (2.9) we can bound the term $\langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{x}^*\rangle$ from below by $f(\mathbf{x}_k) - f(\mathbf{x}^*) = f_k$. Subtracting $f(\mathbf{x}^*)$ on both sides of the above inequality, we arrive at

$$\begin{aligned}\mathbb{E}\left[f_{k+1} \mid (\mathbf{x}_i)_{i=0}^k\right] &\leq (1 + h_k\mathbb{E}[\gamma_k])\, f_k + \frac{h_k^2\mathbb{E}[\gamma_k]L\theta}{2}\left\|\mathbf{x}_k - \mathbf{x}^*\right\|_A^2 + \mathbb{E}[\epsilon_k]\,,\\ &\leq \left(1 + h_k\mathbb{E}[\gamma_k] + \frac{h_k^2\mathbb{E}[\gamma_k]L\theta}{m}\right) f_k + \mathbb{E}[\epsilon_k]\,.\end{aligned}$$

for $h_k \in \mathbb{R}$. The second inequality is due to assumption (3.5). Setting $h_k = -\frac{m}{2L\theta}$ the term in the left bracket becomes $\left(1 - \frac{\mathbb{E}[\gamma_k]m}{4L\theta}\right)$ and the proof continues as the proof of Theorem 3.6. □

First of all, we note that this technique is not restricted to the function class (3.5). The results of Theorem 3.6 and Theorem 3.7 can be obtained by the same technique, if we assume the stronger sufficient decrease (D2) instead of only (D1). The idea to use transformations T_k with $\mathbb{E}[\langle T_k(\mathbf{x}_k), \mathbf{u}_k\rangle\, \mathbf{u}_k] = (\mathbf{x}_k - \mathbf{x}^*)$ was used in [134]. Lemma 3.11 was the key to our results in [239].

To ensure convergence, the second moment $\mathbb{E}[\|\langle T_k(\mathbf{x}_k), \mathbf{u}_k\rangle\, \mathbf{u}_k\|_A^2]$ has to be finite. As its magnitude has a large impact on the convergence rate, one has to provide tight upper bounds. Let us just present one concrete example.

Example 3.13. *Consider the setting of Theorem 3.12 and assume that the search directions $(\mathbf{u}_k)_{k\geq 0}$ are independent copies $\mathbf{u} \sim S_{B^{-1}}^{n-1}$ for $B \in \mathrm{PD}_n$, assume $\mathbf{u}_k$ and γ_k independent. Then $T_k(\mathbf{x}) := nB^{-1}(\mathbf{x} - \mathbf{x}^*)$ satisfies*

$$\mathbb{E}[\langle T_k(\mathbf{x}_k), \mathbf{u}_k\rangle\, \mathbf{u}_k] = n\mathbb{E}[\left\langle B^{-1}(\mathbf{x}_k - \mathbf{x}^*), \mathbf{u}_k\right\rangle \mathbf{u}_k] = \mathbf{x}_k - \mathbf{x}^*\,,$$

and

$$\mathbb{E}\left[\|\langle T_k(\mathbf{x}_k), \mathbf{u}_k\rangle\, \mathbf{u}_k\|_A^2\right] \leq n\kappa_{\mathrm{T}}(AB)\left\|\mathbf{x}_k - \mathbf{x}^*\right\|_A^2\,,$$

where $\kappa_{\mathrm{T}}(AB) := \frac{1}{n+2}\left(\mathrm{Tr}[AB]\lambda_{\min}^{-1}(AB) + 2\right)$.

Proof. The required expected values are presented in Lemma A.7 on page 114 in the appendix. We have $\mathbb{E}[\langle \mathbf{y}, \mathbf{u}_k \rangle \mathbf{u}_k] = \frac{1}{n} B\mathbf{y}$ for $\mathbf{y} \in \mathbb{R}^n$, this shows the first claim for the choice $\mathbf{y} = B^{-1}(\mathbf{x} - \mathbf{x}^*)$. And the second moment $\mathbb{E}[\|n \langle \mathbf{y}, \mathbf{u}_k \rangle \mathbf{u}_k\|_A^2] = \frac{n}{n+2} \left(\mathrm{Tr}[AB] \|\mathbf{y}\|_{B^{-1}}^2 + 2 \|\mathbf{y}\|_A^2 \right)$. It remains to note

$$\frac{n}{n+2} \left(\mathrm{Tr}[AB] \|\mathbf{x}_k - \mathbf{x}^*\|_{B^{-1}}^2 + 2 \|\mathbf{x}_k - \mathbf{x}^*\|_A^2 \right) \leq n \kappa_{\mathrm{T}}(AB) \|\mathbf{x}_k - \mathbf{x}^*\|_A^2 ,$$

where we used Lemma 2.2 from page 35 to bound $\|\mathbf{x}_k - \mathbf{x}^*\|_{B^{-1}}^2 \leq \lambda_{\min}^{-1}(AB) \|\mathbf{x}_k - \mathbf{x}^*\|_A^2$. □

3.6 Two Concentration Bounds

In Section 3.3 above we derived two upper bounds on the expected error $\mathbb{E}[f_N]$ for randomized schemes. We already mentioned that we could also use Theorem 3.3 directly to bound $\mathbb{E}[f_N]$. However, for this we need to calculate the expected values $E[e^{-S_N}]$ and $\mathbb{E}[1/S_N]$ according to part (i) and (ii), respectively. We now argue why it is typically enough to consider only lower bounds on the considerably simpler expression $\mathbb{E}[S_N]$.

Let $s > 0$ and suppose that the inequality $\frac{1}{N} S_N \geq s$ holds for one specific realization of the random variables $\gamma_k \beta_k^2$. For the same realization it holds $S_N \geq Ns$ and therefore $e^{-S_N} \leq e^{-Ns}$ and $1/S_N \leq 1/(Ns)$. If the inequality $\frac{1}{N} S_N \geq s$ holds with high probability, then Theorem 3.3 provides high probability upper bounds on f_N. If the random variables $\gamma_k \beta_k^2$ are independent, then by the Central Limit Theorem the mean $\frac{1}{N} S_N$ is indeed concentrated around its expected value for large N. Thus for s slightly smaller value than $\frac{1}{N} E[S_N]$ and N large enough, the inequality $\frac{1}{N} S_N \geq s$ holds with high probability. Independence of the $\gamma_k \beta_k^2$ does for instance hold for Random Pursuit algorithms if the search directions $(\mathbf{u}_k)_{k \geq 0}$ are generated independently at random and the search steps are then computed as to satisfy (D1) for some fixed gain $\gamma > 0$, even independent of k, say.

Fact 3.14 (Central Limit Theorem)**.** *Let $(X_k)_{k \geq 0}$ be a sequence of independent random variables with finite expectations and variances:*

$$\mu \leq \mathbb{E}[X_k] =: \mu_k \leq \bar{\mu} < \infty , \quad \mathbb{V}[X_k] =: \sigma_k^2 \leq \sigma^2 < \infty , \quad \forall\ k \geq 0. \tag{3.7}$$

The random variables $Z_m := \frac{1}{z_m}\sum_{k=0}^{m-1}(X_k - \mu_k)$, *with* $z_m^2 := \sum_{k=0}^{m-1}\sigma_k^2$, *converge in distribution to a standard normal random variable* $\mathcal{N}(0,1)$ *as* $(m \to \infty)$.

Proof. The random variables (3.7) satisfy Lindenberg's condition [144] and therefore the Central Limit Theorem (see e.g. [9, §7]) holds. □

The bounded[1] random variables $\gamma_k\beta_k^2 \in [0,1]$ clearly satisfy the conditions for the Central Limit Theorem 3.14. Unfortunately, the Central Limit Theorem does not make a statement about the rate of convergence, i.e. what "N large enough" really means. In the following, we provide two simple concentration bounds that answer this question more precisely.

3.6.1 Linear Convergence

Consider the situation in part (i) of Theorem 3.3 and suppose $\mathbb{E}[\gamma_k\beta_k^2] \geq \mu > 0$ for $k \geq 0$ and $C_N = 0$. By independence,

$$\mathbb{E}[f_N] \leq f_0 \prod_{i=0}^{N-1}\left(1 - \frac{m\mathbb{E}[\gamma_k\beta_k^2]}{L}\right) \leq f_0\left(1 - \frac{m\mu}{L}\right)^N . \tag{3.8}$$

The expected value of f_N decreases exponentially. This is also called linear convergence. This decay is fast enough that Markov's inequality gives a useful concentration bound.

Lemma 3.15 (Markov)**.** *Let* $(X_k)_{k\geq 0}$ *be a sequence of nonnegative random variables such that* $\mathbb{E}[X_k] \leq C(1-c)^k$, *for* $0 < c \leq 1$, *and* $C \geq 0$, *and let* $a \geq 1$. *Then for all* $k \geq 0$,

$$X_k \leq aE[X_k] = aC(1-c)^k ,$$

with probability at least $1 - \frac{1}{a}$.

Proof. By the Markov inequality[2], the probability that X_k exceeds its expectation by more than a factor of a is at most $1/a$, and this yields the claim. □

Put differently, for $\epsilon > 0$ we need $K = \frac{1}{c}\ln\frac{C}{\epsilon}$ iterations for $E[X_K] \leq \epsilon$. At the expense of additional $\frac{1}{c}\ln a$ iterations, $X_{K'} \leq \epsilon$ with probability $1 - \frac{1}{a}$, for $K' = \frac{1}{c}\ln\frac{aC}{\epsilon}$.

[1] $\mathbb{V}[X] \leq \frac{(b-a)^2}{4}$ for random variable $X \in [a,b]$, for $-\infty < a \leq b < \infty$, cf. [39].
[2] $\mathbf{Pr}[X \geq a] \leq \mathbb{E}[X]a^{-1}$ for $a > 0$ and nonnegative random variable X.

3.6.2 Small Deviation

To apply Markov's inequality we only need to know the expectation of a random variable. If we have a bound on the variance, we can also apply Chebyshev's Inequality[3].

Lemma 3.16 (Chebyshev)**.** *Let $(X_k)_{k\geq 0}$ be a sequence of independent random variables satisfying* (3.7)*. And denote $S_N := \sum_{k=0}^{N-1} X_k$. Then for $\epsilon > 0$,*

$$\mathbf{Pr}\left[\tfrac{1}{N}S_N \leq \mu - \epsilon\right] \leq \frac{\sigma^2}{N\epsilon^2}\,. \tag{3.9}$$

Proof. The random variable $\frac{1}{N}S_N$, satisfies $\mu \leq \frac{1}{N}\mathbb{E}[S_N] < \infty$ and $\mathbb{V}[\frac{1}{N}S_N] \leq \frac{\sigma^2}{N}$. Therefore, a direct application of Chebyshev's Inequality yields

$$\mathbf{Pr}\left[\tfrac{1}{N}S_N \leq \mu - \epsilon\right] \leq \mathbf{Pr}\left[\tfrac{1}{N}\left|S_N - \mathbb{E}[S_N]\right| \geq \epsilon\right] \leq \frac{\sigma^2}{N\epsilon^2}\,. \qquad \square$$

The ϵ in in (3.9) has to be chosen smaller than μ, say $\epsilon = b\mu$ for $0 < b \leq 1$. In order that $\frac{1}{N}S_N \geq (1-b)\mu$ with nontrivial probability, the number of iterations N has to be at least $N = \Omega(\frac{\sigma^2}{\mu^2})$. This condition simplifies to $N = \Omega(\frac{1}{\mu})$ by the following observation.

Fact 3.17. *Let X be a random variable $X \in [0,1]$. Then $\mathbb{V}[X] \leq \mathbb{E}[X]$.*

Proof. For all $x \in [0,1]$, $x^2 \leq x$, thus $\mathbb{V}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2 \leq \mathbb{E}[X]$. $\square$

For typical applications[4] we can indeed expect $N = \Omega(\frac{1}{\mu})$ and the (high) probability bound of Lemma 3.16 holds.

Remark 3.18. *For typical applications, $N = \Omega(\frac{1}{\mu})$.*

Proof. We observe in Theorem 3.3 that the fastest convergence is obtained in part (i). Although the theorem does not provide a lower bound on the number of iterations N, we conclude with regard to (3.8) that the number N of iterations should be at least $N = \Omega(\frac{1}{\mu})$ in order guarantee a decrease of the initial error f_0 by a constant factor. $\square$

[3] $\mathbf{Pr}[|X - \mathbb{E}[X]| \geq a] \leq \mathbb{V}[X]a^{-2}$ for $a > 0$ and random variable X with $\mathbb{E}[X] < \infty$ and $\mathbb{V}[X] < \infty$.

[4] In [236, 239] $N = \Omega(n)$, and even $N = \Omega(n^2)$ in [233, 235].

Chapter 4

Random Pursuit

In this chapter we are concerned with the question how an algorithmic scheme can generate steps that satisfy the sufficient decrease condition (D1). We study Random Pursuit algorithms where the search directions $(\mathbf{u}_k)_{k\geq 0}$ are independent copies of a random variable $\mathbf{u} \sim \pi$, for specific, but fixed probability distributions. We assume that the step sizes σ_k in scheme (2.2) are generated such as to satisfy condition (D1). We have already mentioned that the step sizes could (for instance) be generated by a line search, and the scheme looks in this case as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathtt{LS}^f(\mathbf{x}_k, \mathbf{u}_k) \cdot \mathbf{u}_k \,. \tag{4.1}$$

The use of a line search oracle has two prominent advantages: (i) it renders the scheme (4.1) invariant under monotonic transformations of the objective functions. By this me mean that the steps taken by (4.1) are the same on any two functions whose level sets agree. Like for the Evolution Strategies discussed in Section 2.4. And (ii) it does not require an additional input, e.g. user defined parameters. The gradient-based schemes, see e.g. [184] typically require an upper bound on the curvature L as an input (however, this parameter can also be estimated).

We have seen in Lemma 3.10 that the exact line search oracle satisfies (D1) with parameter $\gamma = 1$ (and $\epsilon = 0$). The inexact line search oracle (D2) was defined in such a way that that the parameters γ in (D1) and (D2) both agree. However, it might not be clear how a practical line search oracle could be implemented such as to indeed satisfy this condition. We clarify this in Section 4.1 below.

The second ingredient in scheme (4.1) is the probability distribu-

description	$\mathbf{u} \sim \pi$	$\mathbb{E}\big[\beta_A^2(\nabla f(\mathbf{x}), \mathbf{u})\big] \geq (\cdots)^{-1}$	Ex.
Gradient	$\mathbf{u} = \nabla f(\mathbf{x})$	$\frac{(\lambda_{\min}(A)+\lambda_{\max}(A))^2}{4\lambda_{\min}(A)\lambda_{\max}(A)}$	4.6
Newton	$\mathbf{u} = A^{-1}\nabla f(\mathbf{x})$	1	4.7
unit sphere	$\mathbf{u} \sim S_{B^{-1}}^{n-1}$	$n\kappa(AB)$	4.9
		$\frac{n\left(\mathrm{Tr}[AB]\lambda_{\min}^{-1}(AB)+2\right)}{n+2} =: n\kappa_{\mathrm{T}}(AB)$	4.12
$\kappa_{\mathrm{E}} \leq \kappa_{\mathrm{T}} \leq \kappa$	(Rem. 4.13)	$\frac{n(\mathrm{Tr}[AB]\sigma(\nabla f(\mathbf{x}))+2)}{n+2} =: n\kappa_{\mathrm{E}}(A, B, \nabla f(\mathbf{x}))$	4.12
rank-one	$U \sim S_1^{n^2-1}$	$\frac{n(n+2)}{2}$	4.16
discrete	$\mathbf{u} \sim \{\mathbf{e}_i\}_{i=1}^n$	$n\kappa(A)$	4.14
weighted	$\mathbf{u} \sim_w T$	$\left(\left\|C^{-1}\right\|_2 \cdot \|C\|_F\right)^2 =: \kappa_{\mathrm{F}}^2(C)$	4.15
subsample	$\mathbf{u} \sim \{\mathbf{u}_i \sim S^{n-1}\}_{i=1}^{\Theta(n)}$	$(1-\epsilon)n\kappa_T(I_n)$	4.18

Table 4.1: Upper bounds on $\left(\mathbb{E}\big[\beta_A^2\big]\right)^{-1}$ (i.e., lower bounds for $\mathbb{E}\big[\beta_A^2\big]$) for different probability distributions π. The right column denotes the number of the example where the according result is derived and proper notation introduced. The results can be found on page 62–69.

tion π, according to which the search directions are sampled. The most important example that we are going to study are vectors sampled uniformly from a unit sphere S_B^{n-1}, where $B \in \mathrm{PD}_n$ is an arbitrary metric. This should be read as follows: the metric $A \in \mathrm{PD}_n$, that appears for instance in Theorem 3.6, is a purely theoretical variable that we can tune to fit the objective function: it has to hold $f \in C_{m,L}^1(A)$ in metric A for certain parameters m, L that, ideally, are as close as possible. For instance for $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_A^2$ quadratic, we would simple take A itself as the metric, and then $f \in C_{1,1}^1(A)$. However, for a black-box optimization problem (OPT), we don't know the right metric, even if f is quadratic, say. Thus, we have to decide on a metric $B \in \mathrm{PD}_n$ to work in, often we simply take $B = I_n$. We have to quantify the impact on the convergence rate if a suboptimal B is picked.

In Section 4.2 we present a large selection of (well-chosen) examples of probability distributions. Besides the uniform distributions on spheres, we also discuss discrete distributions on either fixed, or randomly generated, sets of (unit) vectors. Lastly, we consider also a specific distribution in SYM_n, the space of symmetric $n \times n$ matrices. All

these three examples will be important for the applications that we consider in Chapter 5.

With regard to Theorem 3.6 we have to study the convergence factor

$$\left(1 - \frac{\mathbb{E}\left[\gamma\beta_A^2\right] m}{L}\right) \tag{4.2}$$

in order to estimate the expected convergence on $f \in C^1_{m,L}(A)$. The quantities m and L are just fixed parameters. If step sizes are generated by a line search, like in (4.1), then it is reasonable to assume that the line search oracle works independently of the current search direction $\mathbf{u}_k$, and that the sufficient decrease can be estimated by two parameters γ and ϵ, uniformly over all steps $k \geq 0$. Hence, all that is left to do is to study the expectation $\mathbb{E}\left[\beta_A^2\right]$ in (4.2). The angle measure $\beta_k = \beta_A(\nabla f(\mathbf{x}_k), \mathbf{u}_k)$ in (A1) was defined as a function of the gradient $\nabla f(\mathbf{x}_k)$ of the objective function at $\mathbf{x}_k$, the search direction $\mathbf{u}_k$ and a positive definite matrix A. In Section 4.2 we study (the expectation of) $\beta_A^2(\nabla f(\mathbf{x}), \mathbf{u})$ for arbitrary $\mathbf{x} \in \mathbb{R}^n$ and search direction $\mathbf{u} \sim \pi$.

4.1 Line Search

From a theoretical point of view, the inexact line search oracle (D2) is quite satisfactory, as it handles relative (γ) and absolute (ϵ) errors. However, from a practical point of view, it is not clear how one can efficiently check if condition (D2) holds, as it involves the unknown quantity $\tilde{f}$. We now define an inexact line search that measures the quality of the step in terms of the step size $\left|\mathtt{LS}^f\right|$ instead of the improvement in function value (D2). Such a condition might be easier to verify, as we will elaborate below. The following definition generalizes the situation discussed in [239].

Definition 4.1 (Relative/absolute accuracy)**.** *Let $\mathbf{x} \in \mathbb{R}^n$, $f \in C^1$ convex, direction $\mathbf{u} \in \mathbb{R}^n$, and parameter $\mu \geq 0$. Denote*

$$\omega_1 := \min\left(\arg\min_h f(\mathbf{x} + h\mathbf{u})\right), \quad \omega_2 := \max\left(\arg\min_h f(\mathbf{x} + h\mathbf{u})\right).$$

That is, the interval $[\omega_1, \omega_2]$ describes the set of line search minima.

(i) A function $\widehat{\mathtt{LS}}^f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a line search oracle with absolute

error μ with respect to normalization $B^{-1} \in \mathrm{PD}_n$, if for $\mathbf{u} \in S^{n-1}_{B^{-1}}$,

$$\omega_1 - \mu \leq \widehat{\mathtt{LS}}^f(\mathbf{x}, \mathbf{u}) \leq \omega_2 + \mu\,. \tag{L1}$$

(ii) A function $\widehat{\mathtt{LS}}^f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a line search oracle with relative error μ, if:

$$\begin{aligned} \omega_1 \leq \widehat{\mathtt{LS}}^f(\mathbf{x}, \mathbf{u}) \leq \omega_2\,, &\qquad \text{if } \omega_1\omega_2 \leq 0\,, \\ \mu \cdot \omega_1 \leq \widehat{\mathtt{LS}}^f(\mathbf{x}, \mathbf{u}) \leq \omega_2 + \delta_f\,, &\qquad \text{if } \omega_1 > 0\,, \\ \mu \cdot \omega_2 \geq \widehat{\mathtt{LS}}^f(\mathbf{x}, \mathbf{u}) \geq \omega_1 - \delta_f\,, &\qquad \text{if } \omega_2 < 0\,, \end{aligned} \tag{L2}$$

where $\delta_f = \sqrt{2\mu(f(\mathbf{x}) - \tilde{f})/L}\,/\,\|\mathbf{u}\|_A$ for $f \in C^1_L(A)$, and $\delta_f = 0$ if $f \notin C^1_L(A)$, i.e. if the curvature of f is not bounded.

Of course, the quantity δ_f in (L2) is unknown in general, which renders this definition less practical than advertised. This problem dependence can be avoided by requiring (L2) to hold for $\delta_f = 0$. The more general condition just shows that (in theory) less accuracy is needed if $f \in C^1_L(A)$.

Lemma 4.2 (L1,L2) $\Rightarrow$ (D2)**.** *Let $\mathbf{x} \in \mathbb{R}^n$, $f \in C^1_L(A)$ for metric $A \in \mathrm{PD}_n$, search direction $\mathbf{u} \in \mathbb{R}^n$, and $\widehat{\mathtt{LS}}^f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ a line search oracle with absolute or relative error.*

(i) If $\widehat{\mathtt{LS}}^f$ satisfies (L1) *for $\mu \geq 0$, then $\mathbf{x}_+ := \mathbf{x} + \widehat{\mathtt{LS}}^f(\mathbf{x}, \mathbf{u})\mathbf{u})$ satisfies the sufficient decrease condition* (D2) *with $\gamma = 1$, $\epsilon = \mu L/2$.*

(ii) If $\widehat{\mathtt{LS}}^f$ satisfies (L2) *for $\mu \geq 0$, then $\mathbf{x}_+ := \mathbf{x} + \widehat{\mathtt{LS}}^f(\mathbf{x}, \mathbf{u})\mathbf{u})$ satisfies the sufficient decrease condition* (D2) *with $\gamma = 1 - \mu$, $\epsilon = 0$.*

Proof. If $\omega_1 \leq \mathbf{x}_+ \leq \omega_2$ then (D2) holds for $\gamma = 1$ and $\epsilon = 0$ by definition (L1, L2) of the line search.

We first show (ii). Assume $\omega_1 \geq 0$ (otherwise replace $\mathbf{u}$ by $-\mathbf{u}$). If $\mu\omega_1 \leq \widehat{\mathtt{LS}}^f \leq \omega_1$ then (D2) follows for $\gamma = (1 - \mu)$ by the definition of convexity (2.9). For $\omega_2 \leq \widehat{\mathtt{LS}}^f \leq \omega_2 + \delta_f$, we use the quadratic upper bound (2.10) and deduce:

$$f(\mathbf{x}_+) \leq \tilde{f} + \frac{L}{2}\left\|\widehat{\mathtt{LS}}^f(\mathbf{x}, \mathbf{u})\mathbf{u} - \omega_2\mathbf{u}\right\|^2_A \leq \tilde{f} + \frac{L\delta_f^2}{2}\|\mathbf{u}\|^2_A\,, \tag{4.3}$$

and by definition of δ_f the right hand side equals (D2) with $\gamma = (1-\mu)$ and $\epsilon = 0$.

Now we proceed to part (i). Assume $\omega_2 \leq \widehat{\mathtt{LS}}^f \leq \omega_2 + \mu$. Using the quadratic upper bound (2.10) we can derive an upper bound on $f(\mathbf{x}_+)$: just replace δ_f by μ in (4.3). With Lemma 2.2 we can continue

$$f(\mathbf{x}_+) \leq \tilde{f} + \frac{L\mu^2}{2} \|\mathbf{u}\|_A^2 \leq \tilde{f} + \frac{L\mu^2 \lambda_{\max}(AB)}{2} \|\mathbf{u}\|_{B^{-1}}^2 ,$$

and we see that (D2) hold for $\gamma = 1$ and $\epsilon = L\mu^2\lambda_{\max}(AB)/2$. □

4.1.1 Bisection

The one dimensional optimization problem (D2) can for instance be solved by Bisection Search. If an accuracy μ in (L1) is fixed, then the approximate localization of $\widehat{\mathtt{LS}}$ can be done with $O(\ln R + \ln \mu^{-1})$ function evaluations. Here R is an upper bound on the largest possible value of $\widehat{\mathtt{LS}}$, for instance R as defined in Theorem 3.3 [115, 130].

4.1.2 Gradient Oracles

The estimation of directional derivatives can sometimes be easier than the estimation of the whole gradient. If the step sizes are chosen proportional to the directional derivative, then (D1) holds.

Example 4.3. *Let $f \in C_L^1(A)$ for metric $A \in \mathrm{PD}_n$. Let $\mathbf{x} \in \mathbb{R}^n$, search direction $\mathbf{u} \in S_A^{n-1}$, and $0 \leq t \leq \frac{2}{L}$. Then $\mathbf{x}_+ := \mathbf{x} - t\,\langle \nabla f(\mathbf{x}), \mathbf{u}\rangle\,\mathbf{u}$ satisfies* (D1) *with parameters $\gamma = tL\,(2-tL)$ and $\epsilon = 0$. Note that $\min_t tL(2-tL) = 1$, and the minimum is attained for $t^* = \frac{1}{L}$.*

Proof. This is a simple consequence of the quadratic upper bound (2.10). We check

$$\begin{aligned} f(\mathbf{x}_+) &\leq f(\mathbf{x}) - t\,\langle \nabla f(\mathbf{x}), \mathbf{u}\rangle^2 + t^2 \frac{L}{2} \langle \nabla f(\mathbf{x}), \mathbf{u}\rangle^2 \\ &= f(\mathbf{x}) - tL\,(2-tL)\,\frac{\beta_A^2(\nabla f(\mathbf{x}), \mathbf{u})}{2L} \|\nabla f(\mathbf{x})\|_{A^{-1}}^2 . \end{aligned}$$

□

If directional derivatives cannot be computed, one can use estimation by finite differences. However, one has to bound the approximation error. Such analysis has for instance been carried out in [184]. In the present work we focused on gradient-free schemes, therefore we don't present more details here.

4.1.3 Special Case: Quadratic Functions

On quadratic functions $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_A^2$ for $A \in \mathrm{PD}_n$ both the steps generated by the exact line search oracle $\mathtt{LS}^f$ and the directional derivatives do coincide.

Remark 4.4. *Let $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_A^2$ quadratic for $A \in \mathrm{PD}_n$ and let $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{u} \in \mathbb{R}^n$. Then $\mathtt{LS}^f(\mathbf{x}, \mathbf{u}) = -\frac{\langle \nabla f(\mathbf{x}), \mathbf{u} \rangle}{\|\mathbf{u}\|_A^2}$.*

Proof.

$$\begin{aligned}\mathtt{LS}^f(\mathbf{x}, \mathbf{u}) &= \arg\min_h \frac{1}{2}(\mathbf{x} + h\mathbf{u})^T A (\mathbf{x} + h\mathbf{u}) \\ &= \arg\min_h \left(h\mathbf{u}^T A\mathbf{x} + \frac{1}{2}h^2 \mathbf{u}^T A\mathbf{u} \right).\end{aligned}$$

The right hand side is minimized for $-\frac{\langle \nabla f(\mathbf{x}), \mathbf{u} \rangle}{\|\mathbf{u}\|_A^2}$. □

On quadratic functions, the exact line search oracle can be easily computed by interpolation (up to numerical precision).

Example 4.5 (Interpolation)**.** *Let $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_A^2$ quadratic for $A \in \mathrm{PD}_n$ and let $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{u} \in \mathbb{R}^n$. For arbitrary shift $s \in \mathbb{R}$, denote $f_s = f(\mathbf{x} + s\mathbf{u})$, $f_+ := f(\mathbf{x} + (s + \epsilon)\mathbf{u})$, and $f_- := f(\mathbf{x} + (s - \epsilon)\mathbf{u})$ for parameter $\epsilon > 0$. Then*

$$\mathtt{LS}^f(\mathbf{x}, \mathbf{u}) = -\frac{b}{a}, \tag{4.4}$$

for curvature $a = \frac{f_+ - 2f_s + f_-}{\epsilon^2}$ and declivity $b = \frac{f_+ - f_-}{2\epsilon}$.

This is a simple consequence of the fact that f is a quadratic function and can be derived with elementary computations. We present the proof in the appendix on page 118 for completeness. The statement can also be generalized by interpolation at three distinct points $\mathbf{x} + s\mathbf{u}$, $\mathbf{x} + (s + \epsilon_1)\mathbf{u})$, $\mathbf{x} + (s - \epsilon_2)\mathbf{u})$, $\epsilon_1, \epsilon_2 > 0$, $\epsilon_1 \neq \epsilon_2$, on the line defined by $\mathbf{x} + t\mathbf{u}$. Likewise, for any convex polynomial p of degree $2k$, interpolation at $2k + 1$ different points yields the exact value of $\mathtt{LS}^p$.

4.1.4 One-Fifth Success Rule

To conclude this short discussion of line search oracles, we would like to present a comment on the 1/5-th success rule that was introduced in

Section 2.4.1. This scheme can be viewed as a very simplistic (inexact) line search oracle: it samples just one point on the line defined by $\mathbf{x}_k$ and the direction $\mathbf{u}_k$ and with constant probability it finds a point with better function value than $\mathbf{x}_k$. Albeit this is not enough to guarantee fast convergence (see Section 2.3.1), it has been shown (see e.g. [25, 222]) that this scheme is efficient on certain quadratic functions.

The proofs in [23, 24, 111, 209, 220] typically work in two steps: they (i) derive what the optimal step size should be to satisfy (2.3), and then (ii) try to show that an actual implementation of (2.3) produces steps that are actually close to the optimal value.

We now put one small comment that should clarify what we could expect from a rigorous proof to show—at least for quadratic objective functions. Assume $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_A^2$ for $A \in \mathrm{PD}_n$, and choose a random search direction $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, I_n)$, as for instance in CMA-ES. We compute the one step progress of (1+1)-ES. For simplicity, we consider only steps that go in the right direction, i.e. $\langle \nabla f(\mathbf{x}), \mathbf{u}\rangle \geq 0$. Note that not all these steps are necessarily successful (leading to a better function value) and would therefore be rejected by the 1/5-th success rule as defined in (2.6). For a fixed step size σ we have by Taylor expansion for $\mathbf{x}_+ := \mathbf{x} + \sigma\mathbf{u}$:

$$\mathbb{E}\big[f(\mathbf{x}_+) \mid \langle \nabla f(\mathbf{x}), \mathbf{u}\rangle \geq 0\big] = f(\mathbf{x}) - \sigma\mathbb{E}\big[|\langle \nabla f(\mathbf{x}), \mathbf{u}\rangle|\big] + \frac{\sigma^2}{2}\mathbb{E}\big[\|\mathbf{u}\|_A^2\big] .$$

The value of the latter expression is presented in Fact A.2 in the appendix. But we don't have a simple expression for the first expectation on the right hand side. By concentration of measure (cf. e.g. [157]) for normal random variables, the value of the square $\langle \nabla f(\mathbf{x}), \mathbf{u}\rangle^2$ is strongly concentrated around its expectation $\mathbb{E}[\langle \nabla f(\mathbf{x}), \mathbf{u}\rangle^2] = \|\nabla f(\mathbf{x})\|^2$ for dimension n large enough. Thus we can argue that $\mathbb{E}[|\langle \nabla f(\mathbf{x}), \mathbf{u}\rangle|] \approx \|\nabla f(\mathbf{x})\|$, which in turn implies that the progress can be estimated as

$$\begin{aligned} f(\mathbf{x}) - \mathbb{E}\big[f(\mathbf{x}_+) \mid \langle \nabla f(\mathbf{x}), \mathbf{u}\rangle \geq 0\big] &\gtrapprox \sigma\|\nabla f(\mathbf{x})\| - \frac{\sigma^2}{2}\mathrm{Tr}[A] \\ &\gtrapprox \frac{1}{2\mathrm{Tr}[A]}\|\nabla f(\mathbf{x})\|^2 . \end{aligned}$$

The second inequality follows by plugging-in the "optimal" step size $\sigma(\mathbf{x}) \approx \frac{\|\nabla f(\mathbf{x})\|}{\mathrm{Tr}[A]}$ that maximizes the term in the middle. For a quadratic function we have $\nabla f(\mathbf{x}) = A\mathbf{x}$, and together with Lemma 2.2 from page 35 we deduce $\|\nabla f(\mathbf{x})\|^2 \geq \lambda_{\min}(A)\|\mathbf{x}\|_A^2 = 2\lambda_{\min}(A)f(\mathbf{x})$. In

summary, the one step progress for the optimal step size is

$$f(\mathbf{x}) - \mathbb{E}\big[f(\mathbf{x}_+) \mid \langle \nabla f(\mathbf{x}), \mathbf{u}\rangle \geq 0\big] \approx \underbrace{\left(1 - \frac{\lambda_{\min}(A)}{\mathrm{Tr}[A]}\right)}_{\approx(1-(n\kappa_{\mathrm{T}}(A))^{-1})} f(\mathbf{x})\,,$$

Thus, not surprisingly, for optimal step size $\sigma(\mathbf{x})$—despite the fact that the step size is *independent* of the direction $\mathbf{u}$, e.g. fixed—we get approximately the same bound on the one step progress as we will derive for Random Pursuit algorithms with exact line search (see Table 4.1, or Example 4.12 below). Of course, the "$\gtrapprox$" in the above derivation has to be read with care. Thus, we do not claim that the scheme (2.3) works as good as an exact line search, but we cannot be surprised if the expected one step progress (E1) is almost as big (up to a small constant factor) as for schemes with exact line search, i.e. as we would expect for an inexact line search oracle for accuracy $\gamma < 1$.

4.2 Search Directions

As mentioned in the introduction, we will now present lower bounds for $\mathbb{E}\big[\beta_k^2\big]$ for certain exemplary sampling distributions. The angle measure β_k in (A1) was defined as a function of the gradient $\nabla f(\mathbf{x}_k)$ of the objective function at $\mathbf{x}_k$, the search direction $\mathbf{u}_k$ and a positive definite matrix A. In the following we study $\beta_A^2(\mathbf{y}, \mathbf{u})$ for arbitrary $\mathbf{y} \in \mathbb{R}^n$ and search direction $\mathbf{u} \in \mathbb{R}^n$. Setting $\mathbf{y} = f(\mathbf{x}_k)$ and $\mathbf{u} = \mathbf{u}_k$ yields β_k^2.

4.2.1 Deterministic Search Directions

Among the popular deterministic schemes, there are essentially two classes: (i) schemes that try to estimate properties of the objective function and compute approximations to the gradient or Newton directions, and (ii) schemes that simply try all (or maybe less) directions from a predefined set and then select the best one. We do not discuss schemes of the second kind now, as their efficiency can easily be obtained by looking at their randomized counterparts (picking a direction uniformly at random from the set).

Example 4.6 (Gradient Direction)**.** *Let $A \in \mathrm{PD}_n$, $\mathbf{y} \in \mathbb{R}^n$, β_A the angle measure as defined in* (A1) *and let $\lambda_{\min/\max}(A)$ denote the smallest and largest eigenvalue of A and $\mathbf{v}_{\min/\max}$ with $\left\|\mathbf{v}_{\min/\max}\right\| = 1$ the*

corresponding eigenvectors. Then

$$\frac{4\lambda_{\min}(A)\lambda_{\max}(A)}{(\lambda_{\min}(A)+\lambda_{\max}(A))^2} \leq \beta_A^2(\mathbf{y},\mathbf{y}) \leq 1\,.$$

The lower bound is obtained if $\mathbf{y}$ *is proportional to either* $(\mathbf{v}_{\min}+\mathbf{v}_{\max})$ *or* $(\mathbf{v}_{\min}-\mathbf{v}_{\max})$*, the upper bound is obtained if* $\mathbf{y}$ *is an eigenvector of* A*.*

Proof. The upper bound is obtained from the Cauchy-Schwarz Inequality[1]. Let $A^{1/2}$ denote the symmetric, positive definite square root of A. Then $\left\langle A^{1/2}\mathbf{y}, A^{-1/2}\mathbf{y}\right\rangle^2 \leq \left\langle A^{1/2}\mathbf{y}, A^{1/2}\mathbf{y}\right\rangle \left\langle A^{-1/2}\mathbf{y}, A^{-1/2}\mathbf{y}\right\rangle$. This shows the upper bound. If $\mathbf{y}$ is an eigenvalue of A with eigenvalue λ, then $\beta_A(\mathbf{y},\mathbf{y}) = \lambda\lambda^{-1} = 1$, and the upper bound is tight. The lower bound is known as Kantorovich's Inequality[2] [204, §20.3], a nice proof can be found in [39] and is omitted here. □

For the choice $\mathbf{y} = \nabla f(\mathbf{x}_k)$, this example shows that the gradient direction is in general not optimal (with respect to the angle condition). A better direction is the Newton-Raphson direction, defined as $A^{-1}\nabla f(\mathbf{x}_k)$.

Example 4.7 (Newton-Raphson Direction)**.** *The direction* $A^{-1}\mathbf{y}$ *is optimal:* $\beta_A(\mathbf{y}, A^{-1}\mathbf{y}) = 1$*.*

4.2.2 Towards Random Search Directions

In the following we study the expression $\beta_A^2(\mathbf{y},\mathbf{u})$ for arbitrary $\mathbf{y} \in \mathbb{R}^n$ and random search direction $\mathbf{u} \in \mathbb{R}^n$. The angle measure β_A is independent of the scaling of $\mathbf{u}$, therefore it suffices to discuss random directions from a compact, centrally symmetric set, like the sphere $S_{B^{-1}}^{n-1} := \{\mathbf{x} \in \mathbb{R}^n \colon \|\mathbf{x}\|_{B^{-1}} = 1\}$ for $B \in \mathrm{PD}_n$.

Lemma 4.8. *Let* $A, B \in \mathrm{PD}_n$ *and* π *a probability distribution over* $S_{B^{-1}}^{n-1}$ *with covariance* $\mathbb{E}_{\mathbf{u}\sim\pi}[\mathbf{u}\mathbf{u}^T] \succeq \Sigma$ *for* $\Sigma \in \mathrm{PD}_n$*. Here* $\succeq$ *denotes the Löwner ordering on the cone of positive semi-definite matrices, and we write* $X \succeq Y$ *if* $X - Y$ *is positive semi-definite. Then*

$$\min_{\mathbf{y}\in\mathbb{R}^n} \mathbb{E}\left[\beta_A^2(\mathbf{y},\mathbf{u})\right] \geq \frac{\lambda_{\min}(A\Sigma)}{\lambda_{\max}(AB)}\,.$$

[1] $|\langle \mathbf{x},\mathbf{y}\rangle|^2 \leq \langle \mathbf{x},\mathbf{x}\rangle \langle \mathbf{y},\mathbf{y}\rangle$ for $\mathbf{x},\mathbf{y} \in \mathbb{R}^n$ and scalar product $\langle\cdot,\cdot\rangle$.

[2] The version here is a special case of Bühler's Inequality [39]: $1 \leq \mathbb{E}[X]\mathbb{E}[1/X] \leq (a+b)^2/(4ab)$, for X random variable on $[a,b]$, $0 < a \leq b < \infty$.

Proof. With Lemma 2.2 we estimate $\|\mathbf{u}\|_A^2 \leq \lambda_{\max}(AB)\,\|\mathbf{u}\|_{B^{-1}}^2 = \lambda_{\max}(AB)$ for $\mathbf{u} \in S_{B^{-1}}^{n-1}$. Therefore by linearity of expectation

$$\lambda_{\max}(AB)\mathbb{E}\left[\beta_A(\mathbf{y},\mathbf{u})\right] \geq \mathbb{E}\left[\frac{\langle \mathbf{y},\mathbf{u}\rangle^2}{\|\mathbf{y}\|_{A^{-1}}^2}\right] = \frac{\mathbf{y}^T\mathbb{E}[\mathbf{u}\mathbf{u}^T]\mathbf{y}}{\|\mathbf{y}\|_{A^{-1}}^2} \geq \frac{\|\mathbf{y}\|_\Sigma^2}{\|\mathbf{y}\|_{A^{-1}}^2}\,.$$

With Lemma 2.2 we have $\|\mathbf{y}\|_\Sigma^2 \geq \lambda_{\min}(A\Sigma)\,\|\mathbf{y}\|_{A^{-1}}^2$ and the statement follows. □

4.2.3 Spherical and Elliptical Distributions

The most important example of a continuous distribution is the uniform distribution on the unit ball S^{n-1}. We consider here a specific class of anisotropic distributions, namely elliptic distributions, see Section 2.5.5.

Example 4.9 (Simple estimate)**.** *Let $A, B \in \mathrm{PD}_n$ and let $\mathbf{u} \sim S_{B^{-1}}^{n-1}$. Then*

$$\min_{\mathbf{y}\in\mathbb{R}^n} \mathbb{E}\left[\beta_A(\mathbf{y},\mathbf{u})\right] \geq \frac{1}{n\kappa(AB)}\,,$$

where $\kappa(AB)$ denotes the condition number of AB.

Proof. By Lemma A.7 we have $\mathbb{E}[\mathbf{u}\mathbf{u}^T] = \frac{1}{n}B$, and the statement follows with Lemma 4.8. □

Remark 4.10 (Beta distribution)**.** *Zielinksy [264] demonstrated that for $B = A^{-1}$, that is, $\mathbf{u} \sim S_A^{n-1}$, the random variable $\beta_A^2(\mathbf{y},\mathbf{u})$ is $\mathrm{Beta}\left(\frac{1}{2}, \frac{n-1}{2}\right)$ distributed (independent of $\mathbf{y} \in \mathbb{R}^n$). Especially, it follows the expression for the variance: $\mathbb{V}[\beta^2(\mathbf{y},\mathbf{u})] = \frac{2(n-1)}{n^2(n+2)}$.*

We see that Example 4.9 provides the best bound for $B = A^{-1}$. For $B = I_n$ the estimate depends on the condition number $\kappa(A)$, and for $B = A$ even worse on $\kappa(A^2)$. The following remark presents a well-known technique that prevents the explosion of the error term and provides a bound in term of only $\kappa(A)$. The matrix C represents the (bad or good) estimate of A^{-1}.

Remark 4.11 (Importance sampling)**.** *Let $A, C \in \mathrm{PD}_n$, $\epsilon > 0$ and let π denote a distribution on S^{n-1} with covariance Σ satisfying $n\Sigma = \epsilon I_n + (1-\epsilon)C$. Then*

$$\min_{\mathbf{y}\in\mathbb{R}^n} \mathbb{E}\left[\beta_A(\mathbf{y},\mathbf{u})\right] \geq \frac{\epsilon}{n\kappa(A)}\,.$$

Proof. Observe $\lambda_{\min}(A\Sigma) = \epsilon\lambda_{\min}(A) + (1-\epsilon)\lambda_{\min}(AC) \geq \epsilon\lambda_{\min}(A)$ and the claim follows from Lemma 4.8. □

However, note that this approach prohibits perfect learning, i.e. the convergence rate cannot become independent of $\kappa(A)$, even if $C = A^{-1}$ (unless $\epsilon \to 0$). If $B = \epsilon I_n + (1-\epsilon)A$ in Example 4.9, then $\kappa(AB^{-1}) = \kappa(BA^{-1}) \geq \epsilon\kappa(A)$.

Example 4.12 (Improved estimates)**.** *Let* $A, B \in \mathrm{PD}_n$ *and let* $\mathbf{u} \sim S^{n-1}_{B^{-1}}$*. Then*

$$\mathbb{E}\left[\beta_A^2(\mathbf{y},\mathbf{u})\right] \geq \frac{n+2}{n\left(\mathrm{Tr}[AB]\sigma(\mathbf{y})+2\right)} =: \frac{1}{n\kappa_{\mathrm{E}}(A,B,\mathbf{y})} \tag{4.5}$$

$$\geq \frac{n+2}{n\left(\mathrm{Tr}[AB]\lambda^{-1}_{\min}(AB)+2\right)} =: \frac{1}{n\kappa_{\mathrm{T}}(AB)}\,, \tag{4.6}$$

where $\sigma(\mathbf{y}) := \|\mathbf{y}\|^2_{(ABA)^{-1}} \cdot \|\mathbf{y}\|^{-2}_{A^{-1}} \leq \lambda^{-1}_{\min}(AB)$*.*

This example improves upon the previous bound $\mathbb{E}\left[\beta_A^2\right] \geq \frac{1}{n\kappa(AB)}$ from Example 4.9: The quantity $\kappa_{\mathrm{T}}(AB)$ depends not only on the two extreme eigenvalues of the matrix AB, but on the trace $\mathrm{Tr}[AB]$. For n large enough, we have $n\kappa_{\mathrm{T}}(AB) \approx \lambda^{-1}_{\min}(AB) \cdot \mathrm{Tr}[AB]$, which could be much smaller than $n\kappa(AB)$, if a many eigenvalues of AB are small.

Remark 4.13. *For* $A, B \in \mathrm{PD}_n$ *and* $\mathbf{y} \in \mathbb{R}^n$ *arbitrary,*

$$\kappa_{\mathrm{E}}(A,B,\mathbf{y}) \leq \kappa_{\mathrm{T}}(AB) \leq \frac{1}{n}\mathrm{Tr}[AB]\lambda^{-1}_{\min}(AB) \leq \kappa(AB)\,.$$

Proof. The first inequality follows directly from Example 4.12. For $AB \in \mathrm{PD}_n$, the latter two follow from Lemma 2.1 on page 34. We observe that the three quantities $\kappa_{\mathrm{T}}(AB)$, $\mathrm{Tr}[AB]\lambda^{-1}_{\min}(AB)$ and $\kappa(AB)$ only depend on the eigenvalues of AB. Therefore the claim follows from the fact that the two matrices AB and $B^{1/2}AB^{1/2} \in \mathrm{PD}_n$ both have the same eigenvalues, see e.g. [204, Prop. 13.2]. □

Proof of Example 4.12. To compute $\mathbb{E}\left[\beta_A^2(\mathbf{y},\mathbf{u})\right]$ we face the challenge to compute the expectation of a ratio of two quadratic forms, $\mathbb{E}\left[\frac{\mathbf{u}^T Y \mathbf{u}}{\mathbf{u}^T A \mathbf{u}}\right]$, for $Y = \mathbf{y}\mathbf{y}^T$. This task has been treated in the literature (see e.g. [156] or the discussion in Section A.1.3) but the formula depends on integrals which have to be approximated numerically. We are only interested in

a lower bound, therefore we can apply the following trick. For fixed $\mathbf{y} \in \mathbb{R}^n$ and $\mathbf{u} \in \mathbb{R}^n$ we observe

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

where the equality follows by standard calculus, and the inequality by suboptimally setting $t = h \left\langle (AB)^{-1} \mathbf{y}, \mathbf{u} \right\rangle$. The maximum is a convex function. Let denote $E_A = \mathbb{E}\left[\frac{\langle \mathbf{y}, \mathbf{u} \rangle^2}{\mathbf{u}^T A \mathbf{u}} \right]$. With Jensen's inequality[3] and the expectations from Lemma A.7 on page 114 in the appendix, we estimate

$$\begin{aligned}E_A &\geq \max_h \left(2h \mathbb{E}\left[\left\langle (AB)^{-1} \mathbf{y}, \mathbf{u} \right\rangle \langle \mathbf{y}, \mathbf{u} \rangle \right] - h^2 \mathbb{E}\left[\left\| \left\langle (AB)^{-1} \mathbf{y}, \mathbf{u} \right\rangle \mathbf{u} \right\|_A^2 \right] \right) \\ &= \max_h \left(2h \frac{\|\mathbf{y}\|_{A^{-1}}^2}{n} - h^2 \frac{\mathrm{Tr}[AB] \, \|\mathbf{y}\|_{(ABA)^{-1}}^2 + 2 \|\mathbf{y}\|_{A^{-1}}^2}{n(n+2)} \right) \\ &\geq \frac{(n+2) \|\mathbf{y}\|_{A^{-1}}^4}{n\left(\mathrm{Tr}[AB] \, \|\mathbf{y}\|_{(ABA)^{-1}}^2 + 2 \|\mathbf{y}\|_{A^{-1}}^2 \right)} = \frac{\|\mathbf{y}\|_{A^{-1}}^2}{n \kappa_{\mathrm{E}}(A, B, \mathbf{y})} ,\end{aligned}$$

where the last inequality follows by the choice $h = \frac{\|\mathbf{y}\|_{A^{-1}}^2}{\kappa_{\mathrm{E}}(A,B,\mathbf{y})}$. This implies inequality (4.5), as claimed. With Lemma 2.2 from page 35 we estimate $\|\mathbf{y}\|_{(ABA)^{-1}}^2 \leq \|\mathbf{y}\|_{A^{-1}}^2 \lambda_{\max}((BA)^{-1}) = \|\mathbf{y}\|_{A^{-1}}^2 \lambda_{\min}^{-1}(AB)$, and the last inequality (4.6) follows as well. □

4.2.4 Discrete Distributions

The entries of a random unit vector are nonzero with probability one. Thus any computation that involves this vector, say computing a scalar product, requires at least $\Omega(n)$ time. From a computational point of view, sparse vectors (with only constantly many nonzero entries) are preferable if (i) the sparsity can be efficiently exploited, and (ii) the convergence rate is not much worse than for the best possible search directions. We show, that the expected squared angle measure for uniform random standard unit vectors is not worse than for uniform random unit vectors. This idea is exploited for instance for Random Coordinate Descent [185] or also for a similar application in [140].

[3] $\mathbb{E}[\phi(X)] \geq \phi(\mathbb{E}[X])$, for convex ϕ and random variable X.

Example 4.14. *Let $A \in \mathrm{PD}_n$ and let $\mathbf{u} \sim \{\mathbf{e}_i \colon i = 1, \ldots, n\}$, the set of standard unit vectors, $\langle \mathbf{e}_i, \mathbf{x} \rangle = x_i$, for all $\mathbf{x} \in \mathbb{R}^n$. Then*

$$\min_{\mathbf{y} \in \mathbb{R}^n} \mathbb{E}\left[\beta_A^2(\mathbf{y}, \mathbf{u})\right] \geq \frac{1}{n}\,.$$

Proof. We calculate the expectation $\mathbb{E}[\mathbf{u}\mathbf{u}^T]$ explicitly:

$$\mathbb{E}\left[\mathbf{u}\mathbf{u}^T\right] = \frac{1}{n} \sum_{i=1}^{n} \mathbf{e}_i \mathbf{e}_i^T = \frac{1}{n} I_n\,,$$

and the statement follows with Lemma 4.8. □

Now we consider an example of a non-uniform discrete distribution. Suppose $A = B = I_n$ and let $T \subset S^{n-1}$ denote a (discrete) set of m unit vectors in $\mathbb{R}^n$. Instead of sampling the vectors $\mathbf{y}_i \in T$ for $i = 1, \ldots, m$ uniformly, sample each vector $\mathbf{y}_i \in T$ with weight $w_i \geq 0$, denoted as $\mathbf{u} \sim_w T$. We will use this example in Section 5.3 below.

Example 4.15 (Weighted Discrete; adapted from [240])**.** *Let $A = B = I_n$ and $C \in \mathbb{R}^{m \times n}$ with non-zero row vectors $\mathbf{c}_i^T \in \mathbb{R}^n$, for $i = 1, \ldots, m$ and $m \geq n$. Let $T := \{\frac{\mathbf{c}_i}{\|\mathbf{c}_i\|} \colon i = 1, \ldots, m\} \subset S^{n-1}$ denote the normalized set of row vectors and sample $\mathbf{u} \sim_w T$ with weights according to the norm $w_i = \|\mathbf{x}_i\|_2^2$. If C has full rank n, the left inverse C^{-1} with $C^{-1}C = I_n$ exists and*

$$\min_{\mathbf{y} \in \mathbb{R}^n} \mathbb{E}\left[\beta_{I_n}^2(\mathbf{y}, \mathbf{u})\right] \geq \left(\left\|C^{-1}\right\|_2 \|C\|_F\right)^{-2} =: \kappa_{\mathrm{F}}^{-2}(C)\,.$$

The proof of this statement goes in two parts: we want to apply Lemma 4.8 but for this we need a lower bound on the smallest eigenvalue of the covariance Σ, with $\Sigma = \mathbb{E}_{\mathbf{u} \sim_w T}[\mathbf{u}\mathbf{u}^T]$. We provide the required details in the appendix on page 119.

4.2.5 Rank-One Matrices

We would like to present another application in the space SYM_n, of symmetric $n \times n$ matrices. This example will be of importance later in Section 5.2. Let $S^{n^2-1} = \{X \in \mathrm{SYM}_n \colon \|X\|_F = 1\}$ denote the unit norm ball in SYM_n. Ideally, we would like to sample search directions uniformly from S^{n^2-1}. However, we have seen in the previous Section 4.2.4 that sampling from a subset of the norm ball can be sufficient for the optimization task, and such distributions can often more

easily be generated. Indeed, we can define a distribution over symmetric rank-one matrices by setting $U = \mathbf{u}\mathbf{u}^T$ for $\mathbf{u} \sim S^{n-1}$, denoted as $U \sim S_1^{n^2-1}$. For simplicity, we assume that $A = B = I_{n^2}$.

Example 4.16 (Rank-one matrices)**.** *Let* $\beta(Y,U) := \frac{\langle Y,U\rangle}{\|Y\|_F \|U\|_F}$ *denote the angle measure in* SYM_n *and let* $U \sim S_1^{n^2-1}$*. Then*

$$\mathbb{E}\left[\beta^2(Y,U)\right] = \frac{2 + \mathrm{Tr}[Y]^2 \, \|Y\|_F^{-2}}{n(n+2)} \geq \frac{2}{n(n+2)}\,,$$

for all $Y \in \mathrm{SYM}_n$*.*

Proof. The result follows directly from Lemma A.7. □

4.2.6 Sampling from Random Sets

In Example 4.15 we discussed random sampling of search directions from a fixed set of vectors. However, we did not address the question of how to generate a set with good search directions. It turns out that a linear number of *random* vectors from S^{n-1} is a good choice with high probability. This result is not due to us, but follows from a more general Theorem by Adamczak et al. [3].

The problem that we consider (in the form as stated in [249]) is the following: how well can one approximate one dimensional marginals of a distribution in $\mathbb{R}^n$ by sampling? Consider a random variable X and sample N independent copies $(X_i)_{i=0}^N$ of X. The hope is, that the empirical marginal p-th moments of these samples give a good approximation of the actual marginal p-th moment,

$$\sup_{\mathbf{x}\in S^{n-1}} \left| \frac{1}{N} \sum_{i=1}^{N} |\langle X_i, \mathbf{x}\rangle|^p - \mathbb{E}\big[|\langle X, \mathbf{x}\rangle|^p\big] \right| \leq \epsilon\,. \tag{4.7}$$

One is interested in the sample complexity N for which (4.7) holds with high probability. For $p = 2$, this problem is equivalent to approximating the covariance matrix of X by a sample covariance matrix. This question was investigated by Kannan et al. [125] motivated by the problem of computing volumes of convex bodies in high dimensions. If X is standard normal, then $\mathbb{E}[XX^T] = I_n$ and $N = \Omega(n)$ samples are enough [49]. In general, for distributions with $\mathbb{E}[XX^T] = I_n$, $N = \Omega(n \ln n)$ is needed, as can be seen by considering a random vector X uniformly distributed on a set of n orthogonal vectors of

length $\sqrt{n}$ [214]. For distributions over convex bodies Bourgain [31] was the first to show $N = O(n \ln^3 n)$, this was later improved to $N = O(n \ln n)$ [3, 10, 70, 86, 161, 195, 213, 249]. For $p \neq 2$ the problem was also studied in [3, 32, 71, 86, 160]

In our case, we sample vectors uniformly from the unit sphere. The result of Rudelson [213, 214], shows that $N = O(n \ln n)$ are enough to approximate the covariance. The logarithmic term was recently removed by Adamczak et al. [3]. Concretely, their result implies the following:

Fact 4.17 (Adamczak et al. [3])**.** *Let $\mathbf{u}_1, \dots, \mathbf{u}_N$ be i.i.d. copies of $\mathbf{u} \sim S^{n-1}$. For every $0 < \epsilon < 1$ and $t > 1$, there exists $C(\epsilon, t) > 0$, polynomially depending on ϵ and t, such that if $N \geq C(\epsilon, t)n$, then with probability at least $1 - e^{-t\sqrt{n}}$,*

$$\left\| \frac{1}{N} \sum_{i=1}^{N} \mathbf{u}_i \mathbf{u}_i^T - \mathbb{E}\left[\mathbf{u}\mathbf{u}^T\right] \right\|_2 \leq \epsilon \left\| \mathbb{E}\left[\mathbf{u}\mathbf{u}^T\right] \right\|_2 . \tag{4.8}$$

To provide a lower bound on the squared angle measure, we essentially only needed to provide a lower bound on the covariance, see Example 4.9 and 4.14. Following the spirit of this approach, we see that instead of sampling from the unit sphere S^{n-1}, it is enough to sample uniformly from a fixed set $\{\mathbf{u}_i\}_{i=1}^{\Theta(n)}$, consisting of a linear number of independent copies of a random variable $\mathbf{u} \sim S^{n-1}$.

Example 4.18 (Subsample Sphere)**.** *Let $A \in \mathrm{PD}_n$ and let $T = \{\mathbf{v}_i : i = 1, \dots, N\}$ denote a set of N i.i.d. copies of a random unit vector $\mathbf{v} \sim S^{n-1}$. For $N \geq C(\epsilon, t)n$, with ϵ, t and $C(\epsilon, t)$ as in Fact 4.17, then with probability at least $1 - e^{-t\sqrt{n}}$,*

$$\min_{\mathbf{y} \in \mathbb{R}^n} \mathbb{E}_{\mathbf{u} \sim T} \left[\beta_A^2(\mathbf{y}, \mathbf{u})\right] \geq \frac{1 - \epsilon}{n\kappa(A)} .$$

For the important case of $S_1^{n^2-1}$ distributed rank-one matrices, we cannot just rely on Fact 4.17, but we have to refer to the genuine result from Adamczak et al. [3, Thm. 4.2]. A sketch of how this theorem has to be applied can be found in the appendix on page 120.

Example 4.19 (Subsample Rank-One Matrices)**.** *Let $T = \{V_i : i = 1, \dots, N\}$ denote a set of N i.i.d. copies of a random rank-one matrices $V \sim S_1^{n^2-1}$. For $N \geq C(\epsilon, t)n^2$, with ϵ, t and $C(\epsilon, t)$ according to [3, Thm. 4.2], then with probability at least $1 - e^{-t\sqrt{n}}$,*

$$\min_{Y \in \mathrm{SYM}_n} \mathbb{E}_{U \sim T} \left[\beta^2(Y, U)\right] \geq \frac{2(1 - \epsilon)}{n(n + 2)} .$$

4.3 Discussion

We have seen many examples. To conclude this chapter, we discuss some of these results. We focus on our results obtained for uniformly chosen vectors from a unit sphere. The other sampling distributions will prove their use later in Chapter 5.

In Section 4.3.1 below, we compare the bounds which depend on κ_{E}, κ_{T} and κ (see Table 4.1). In Section 4.3.2 we discuss the special case of the Random Pursuit algorithm (4.1) with exact line search applied to quadratic functions.

4.3.1 Summary of Selected Results

We compare the bounds on the convergence rate from Examples 4.9 and 4.12. For this, we consider three different quadratic functions. We do not only compare the theoretical bounds on the convergence rate, but we also check if these bounds can indeed describe the practically observed behavior of Random Pursuit.

We consider the following setting: we compare the derived bounds for a Random Pursuit algorithm (4.1) with exact line search that samples the search directions uniformly at random from the standard unit sphere $S^{n-1} = S_{I_n}^{n-1}$. The exact line search oracle satisfies (D2) with parameters $\gamma = 1$ and $\epsilon = 0$. Every quadratic function f is in $C^1_{1,1}(A)$, for some metric $A = \nabla^2 f$, the Hessian matrix. Therefore, the convergence factor encountered in Theorem 3.6 and repeated in (4.2) at the beginning of this chapter, reduces in our setting to

$$\left(1 - \mathbb{E}\left[\beta_A^2(\nabla f(\mathbf{x}), \mathbf{u})\right]\right) ,$$

and only depends on the value $\mathbb{E}\left[\beta_A^2(\nabla f(\mathbf{x}), \mathbf{u})\right]$ that we have calculated in Section 4.2.

Simple vs. Improved Bounds

Consider the three quadratic functions f_{two}, f_{flat} and f_{exp}, listed in Table 2.3. They have, i.e. their Hessian matrices, all the same condition number. Therefore, the upper bound on the convergence factor derived in Example 4.9 is identical for all functions:

$$\left(1 - \frac{1}{n\kappa}\right) = \left(1 - \frac{1}{nL}\right) . \tag{4.9}$$

The improved bound from Example 4.12 depends on the trace of the eigenvalue spectrum of the Hessian matrices. We see in Table 2.3 that the two functions f_{two} and f_{flat} have the same trace, and the factor derived in Example 4.12 reads for these two functions as

$$\left(1 - \frac{1}{n\kappa_{\text{T}}}\right) = \left(1 - \frac{(n+2)}{n(n(L+1)/2+2)}\right) \approx \left(1 - \frac{2}{n\kappa}\right), \qquad (4.10)$$

that is, an improvement by a factor of roughly 2 compared to (4.9). The trace of the Hessian of f_{exp} is smaller and we have even a better bound. For $L = 1\text{E}6$ and dimension $n = 20$, say, the trace is roughly 4 times smaller than the trace of f_{two}, and consequently the convergence factor can be estimated as $\left(1 - \frac{1}{n\kappa_{\text{T}}(f_{\text{exp}})}\right) \approx \left(1 - \frac{8}{n\kappa}\right)$.

The two functions f_{two} and f_{flat} can only be distinguished by means of the third quantity κ_{E}, which was also derived in Example 4.12. We will discuss κ_{E} in the next subsection. For now, we only observe that $\kappa_{\text{E}}\left(f_{\text{two}}, I_n, \mathbf{x}_0\right) = \kappa_{\text{E}}\left(f_{\text{flat}}, I_n, \mathbf{x}_0\right)$ for $\mathbf{x}_0 = \mathbf{e}_1$. Hence, we use this point as a starting point for the numerical illustration.

Figure 4.1 depicts the theoretically derived bounds $\left(1 - \frac{1}{n\kappa}\right)$ (which is the same for all functions), and $\left(1 - \frac{1}{n\kappa_{\text{T}}}\right)$, which is the same for f_{two} and f_{flat} and slightly better for f_{exp}. We see, that the empirically observed behavior of Random Pursuit on these functions is (i) always better than predicted by the upper bounds, but (ii) the performance on f_{two} and f_{flat} is roughly identical (slightly better on f_{two}) as we would predict from the factor (4.10).

The Exact Convergence Factor

In Example 4.12 we also derived a even better bound on the convergence rate, the expression (4.5) depending on κ_{E}. It depends not only on the trace of Hessian matrices, but also on the position $\mathbf{x} \in \mathbb{R}^n$, an thus takes the full eigenvalue spectrum into account. In Figure 4.1 we started the numerical experiments at $\mathbf{x}_0 = \mathbf{e}_1$. For this position, $\kappa_{\text{E}}(f, I_n, \mathbf{x}_0) = \kappa_{\text{T}}(f)$, for all three functions. Thus κ_{E} does not admit any (theoretical) improvement on the convergence rate. Intuitively this matches our intuition: the level sets of all three functions are stretched out along the x_1-axis, and very "skinny". Hence, once the algorithm gets trapped close to the x_1-axis, the only improvements can be made for search directions almost parallel to the x_1-axis. The algorithm thus "crawls" very slowly along this valley and also the exact line search cannot help to escape this situation.

To demonstrate qualitatively the convergence factor $\left(1 - \frac{1}{n\kappa_{\mathrm{E}}}\right)$, we purposefully start the algorithmic schemes outside this valley. We set $\mathbf{x}_0 = \mathbf{e}_n$. For the functions f_{two} and f_{flat} we estimate

$$\left(1 - \frac{1}{n\kappa_{\mathrm{E}}(f, I_n, \mathbf{x}_0)}\right) = \left(1 - \frac{(n+2)}{n\left(\frac{n(L+1)}{2L} + 2\right)}\right) \approx \left(1 - \frac{2}{n}\right), \quad (4.11)$$

independent of L. As above, the factor for f_{exp} is even slightly better, approximately $(1 - 8/n)$. These bounds therefore predict rapid convergence if the scheme is started at $\mathbf{x}_0 = \mathbf{e}_n$. However, as soon as the scheme gets trapped in the valley, we again expect slower convergence, with the rate predicted by $\left(1 - \frac{1}{n\kappa_{\mathrm{T}}}\right)$.

Figure 4.2 shows the fast convergence at the beginning. However, after no more than 350 iterations the schemes get trapped and converge slower. The convergence rate is best for f_{exp} and approximately the same for f_{two} and f_{flat}, as predicted. The plot at the top shows an interesting phenomena: although the convergence rate after 350 iterations is the best on f_{exp}, the absolute function value is the largest among the three functions.

We see that κ_{E} describes especially the behavior of Random Pursuit at at the beginning, i.e. at "non-typical" search points. After a tune-in phase, the scheme converges with a rate that is very close to the one predicted by the quantity κ_{T}. We observed this tune-in phase already in [239] and [235], but we did not present an explanation.

4.3.2 Viewed from a Different Angle

Let us recite our proof technique. In the presented framework, we can completely decouple the discussion of the step sizes from the search directions. However, one might wonder if one could do significantly better by analyzing both components at the same time. But this seems to be more difficult. In this section, we will derive the one step progress (similar to (E1) on page 47) for a Random Pursuit algorithm (4.1) with exact line search on quadratic functions f. We assume $f \in C^1_{1,1}(A)$ for $A \in \mathrm{PD}_n$, that is, $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2_A$ and that we sample the search directions $\mathbf{u} \sim S^{n-1}$.

Lemma 4.20 (Exact one step progress)**.** *Let* $f(\mathbf{x}) := \frac{1}{2}\|\mathbf{x}\|^2_A$ *quadratic with* $A \in \mathrm{PD}_n$*. Let* $\mathbf{x} \in \mathbb{R}^n$ *and* $\mathbf{u} \sim S^{n-1}$*, and* $\mathbf{x}_+ := \mathbf{x} + \mathtt{LS}^f(\mathbf{x}, \mathbf{u})\mathbf{u}$

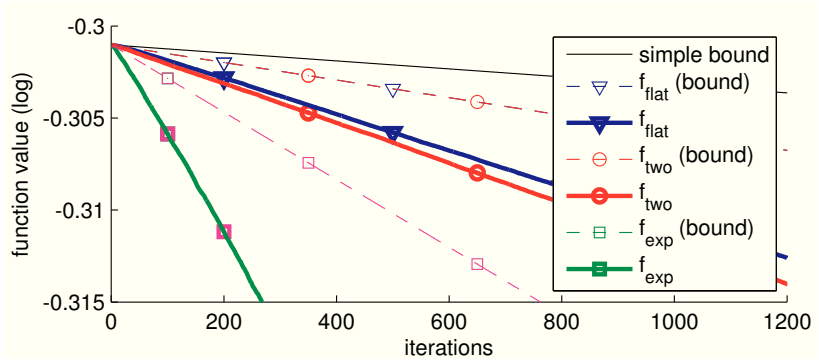

Figure 4.1: Evolution of function value $\log f$ vs. iterations of Random Pursuit with exact line search ($\gamma = 1$) and search directions $\mathbf{u}_k \sim S^{n-1}$; $\mathbf{x}_0 = \mathbf{e}_1$ in $n = 20$ dimensions. Mean of 51 independent runs (bold), and corresponding upper bounds $\left(1 - \frac{1}{n\kappa_{\mathrm{T}}}\right)$ (dashed) from Example 4.12. The solid black line shows the simple upper bound $\left(1 - \frac{1}{n\kappa}\right)$ from Example 4.9.

for exact line search oracle $\mathtt{LS}^f$*. Then*

$$\mathbb{E}\left[f(\mathbf{x}_+) \mid \mathbf{x}\right] = f(\mathbf{x}) - \frac{1}{2}\left\|\nabla f(\mathbf{x})\right\|^2_{Q_A}, \tag{4.12}$$

with $Q_A = \mathbb{E}\left[\frac{\mathbf{u}\mathbf{u}^T}{\|\mathbf{u}\|^2_A}\right]$ *a fixed matrix.*

The proof of this lemma is presented on page 120 in the appendix. However, note that we cannot compute the matrix Q_A of Lemma 4.20 analytically, even if A is known.

Example 4.21. *Let* $A = \ell I_n$ *for* $\ell \geq 0$*. Then* $Q(A) = \frac{1}{n\ell} I_n$*.*

Proof. This claim follows directly from Corollary A.7. □

We now derive two properties of Q_A, the proof of the first one can again be found in the Appendix B.6 on page 120.

Lemma 4.22 ($\lambda_{\min}(AQ_A)$)**.** *For* Q_A *as in the setting of Lemma 4.20,* $\lambda^{-1}_{\min}(AQ_A) \leq n\kappa_{\mathrm{T}}(A)$*.*

Lemma 4.23 (Insufficient[4])**.** *For* Q_A *as in the setting of Lemma 4.20,*

$$\mathbb{E}\left[\left\|\mathtt{LS}^f(\mathbf{x},\mathbf{u})\mathbf{u}\right\|^2_{Q_A^{-1}}\right] \leq \lambda^{-1}_{\min}(AQ_A)\left\|\nabla f(\mathbf{x})\right\|^2_{Q_A} \leq n\kappa_{\mathrm{T}}(A)\left\|\nabla f(\mathbf{x})\right\|^2_{Q_A}.$$

[4]This denotation will become evident by the discussion in Chapter 6, see especially page 96.

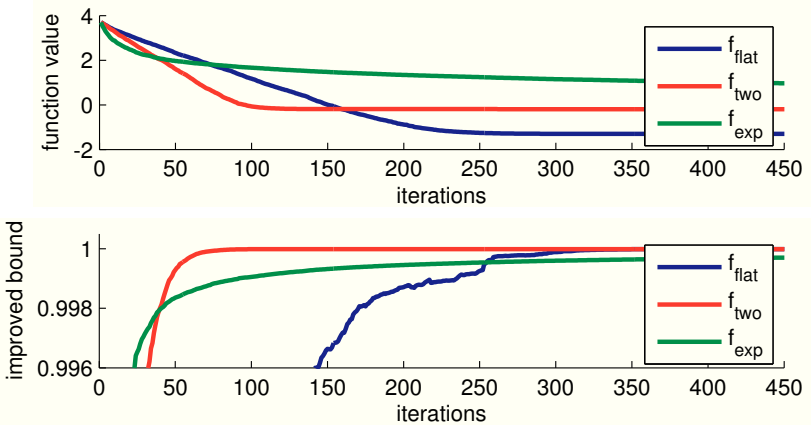

Figure 4.2: Top: Evolution of function value $\log f$ vs. iterations of Random Pursuit with exact line search ($\gamma = 1$) and search directions $\mathbf{u}_k \sim S^{n-1}$; $\mathbf{x}_0 = \mathbf{e}_n$ in $n = 20$ dimensions. Mean of 51 independent runs. Bottom: Mean of the exact bounds $\left(1 - \frac{1}{n\kappa_{\mathrm{E}}}\right)$ from Example 4.12.

Proof of Lemma 4.23. We write out $\mathtt{LS}^f$ explicitly:

$$\left\|\mathtt{LS}^f(\mathbf{x},\mathbf{u})\mathbf{u}\right\|^2_{Q_A^{-1}} = \frac{\|\mathbf{u}\|^2_{Q_A^{-1}}}{\|\mathbf{u}\|^2_A} \cdot \frac{\langle \nabla f(\mathbf{x}), \mathbf{u}\rangle^2}{\|\mathbf{u}\|^2_A} \leq n\kappa_{\mathrm{T}}(A)\frac{\langle \nabla f(\mathbf{x}), \mathbf{u}\rangle^2}{\|\mathbf{u}\|^2_A}\,.$$

The inequality follows by estimation of the first factor. By Lemma 2.2 we have $\|\mathbf{u}\|^2_{Q_A^{-1}} \leq \lambda_{\max}(A^{-1}Q_A^{-1})\,\|\mathbf{u}\|^2_A = \lambda^{-1}_{\min}(AQ_A)\,\|\mathbf{u}\|^2_A$. The minimal eigenvalue of AQ_A was estimated in Lemma 4.22. The claim follows by taking the expectation on both sides and the definition of Q_A. □

Lemma 4.23 bounds the second moment of $\mathtt{LS}^f$ in terms of the Q_A-norm of the gradient $\nabla f(\mathbf{x})$. The pessimistic bound $\|\mathbf{u}\|^2_{Q_A^{-1}} \leq \lambda^{-1}_{\min}(AQ_A)\,\|\mathbf{u}\|^2_A$ holds for any $\mathbf{u} \in \mathbb{R}^n$, however, the bound is attained only for $\mathbf{u}$ parallel to the eigenvector corresponding to the smallest eigenvalue of AQ_A. Here we consider random $\mathbf{u}$, thus we could hope for a much better estimate. Preliminary experiments (for instance on the functions presented in Table 2.3) suggest that an upper bound of the form $cn\,\|\nabla f(\mathbf{x})\|^2_{Q_A}$, where c is an absolute constant, could hold for many functions f, i.e. metrics $A \in \mathrm{PD}_n$.

Du beurre!
Donnez-moi du beurre!
Toujours du beurre!

Fernand Point
a founder of modern French cuisine

Chapter 5

Applications

In this chapter we study three applications of Random Pursuit algorithms. By this we do not mean that we apply one of the algorithms from the previous chapter just to three arbitrary optimization problems. We rather present three problems and show that the typical algorithms that are used to solve these problems are actually Random Pursuit algorithms in disguise. These previously unrelated applications are here presented in a unifying way.

For all three applications, we restrict ourselves to optimization of *quadratic* objective functions. Two of the applications naturally arise in this setting, whereas for the third one, the restriction to quadratic functions was imposed by us, mainly to simplify the presentation—as we will argue below.

We remind the reader that the implementation of an (exact) line search procedure is not an issue on quadratic functions. In Example 4.5 we have demonstrated that an exact line search oracle can be obtained by interpolation of the function values at three distinct points along the search direction. Thus we assume throughout the whole section that the sufficient decrease condition (D2), and therefore also (D1), hold with parameters $\gamma = 1$ and $\epsilon = 0$. Of course, the use of an exact line search is not *required* for the analysis in this section. In case of an inexact line search oracle (like for instance adaptive step size control (2.7)), the gain and error parameters need to be carried along. We now list the three applications that we discuss in this chapter.

System of Linear Equations. We consider the problem of solving a linear system $A\mathbf{x} = \mathbf{b}$, where $A \in \mathbb{R}^{m\times n}$ has full rank n, $m \geq n$, $\mathbf{b} \in \mathbb{R}^m$ and unknown $\mathbf{x} \in \mathbb{R}^n$. This problem can be solved by Gaussian Elimination. However, especially for the over-determined case $m \gg n$, iterative schemes are often not only computationally more efficient, they are also very easy to implement (see e.g. [173]). We demonstrate that the randomized Kaczmarz method [122, 240] is a Random Pursuit algorithm with exact line search and search directions sampled proportional to given weights from a discrete set of directions (cf. Example 4.15). This algorithm can in practice also be applied if the linear system is corrupted by noise, i.e. no solution exists. In this case, it still converges to the solution of the uncorrupted system within an error margin (see [173] and references therein).

Metric Learning. Consider the optimization problem (OPT), for convex objective function $g\colon \mathbb{R}^n \to \mathbb{R}$. Suppose $g \in C^2$ is strongly convex with minimizer $\mathbf{x}^* \in \mathbb{R}^n$. By Taylor expansion we can write

$$g(\mathbf{x}) = g(\mathbf{x}^*) + \frac{1}{2}\left\|\mathbf{x}-\mathbf{x}^*\right\|^2_{\nabla^2 g(\mathbf{x}^*)} + O\left(\left\|\mathbf{x}-\mathbf{x}^*\right\|^3_{I_n}\right) \qquad \text{for } \mathbf{x} \to \mathbf{x}^*\,.$$

We see that g behaves in the neighborhood of $\mathbf{x}^*$ like a quadratic function $f(\mathbf{x}) = \frac{1}{2}\left\|\mathbf{x}\right\|^2_A$ for metric $A = \nabla^2 g(\mathbf{x}^*)$. For simplicity, we will restrict ourselves in the following to optimization of quadratic objective functions. For the convergence results in Chapters 3 we measured the deviation of the objective function from a quadratic function by the parameters m and L, thus here $m = L$.

In Section 4.3 we discussed the convergence factor

$$\varrho_A(B) := 1 - \frac{1}{n\kappa(AB^{-1})}\,. \tag{5.1}$$

that describes the convergence rate of a Random Pursuit algorithm with exact line search on the quadratic function $\frac{1}{2}\left\|\mathbf{x}\right\|^2_A$ if the search directions are sampled from S_B^{n-1} for $B \in \mathrm{PD}_n$. If $B = I_n$, then ϱ_A depends on the condition number of the (unknown) metric A. Suppose that we have a sequence $(B_k)_{k\geq 0}$ of *estimates* of A, that satisfy $B_k \to A$ for $k \to \infty$. But then also $\varrho_A(B_k) \to \left(1-\frac{1}{n}\right)$, and for K large enough, the convergence factor $\varrho_A(B_K) \approx \left(1-\frac{1}{n}\right)$. This observation is the key to variable metric schemes: those schemes comprise algorithmic routines that generate the estimate B_k (or their inverses B_k^{-1}, as it is the case for the CMA-ES presented in Section 2.4). In the scope

of this thesis, we could not find any new theoretical results regarding the update mechanism that is implemented in CMA-ES. A welcome alternative has recently been introduced by Leventhal and Lewis [141], termed Randomized Hessian Estimation scheme (RHE).

Leventhal and Lewis [141] already presented a convergence analysis for (RHE). Our investigation was twofold: we studied different implementations of (RHE) and empirically compared (RHE) with CMA-ES or Gaussian Adaptation [235]. We also studied the evolution of the convergence factor, which turns out to be dependent on the chosen implementation of the scheme. In Section 5.2 we revisit (RHE) and present conclusions of our empirical investigations. Especially, we show that (RHE) can be implemented in such a way that its complexity is independent of the initial approximation error, i.e. $\|B_0 - A\|_F$.

Matrix Valued Random Pursuit. Interestingly, it turns out that (RHE) is a special instance of a Random Pursuit algorithm. The search space is SYM_n, the space of $n \times n$ symmetric matrices, and the objective function is the squared distance of the current estimation B_k to the ground truth A. We analyze this process in a pure theoretical setting in Section 5.1 below, and essentially show that the analysis provided by Leventhal and Lewis [141] is tight, up to a factor of 2.

5.1 Random Pursuit in a Hilbert Space

We consider the sequence of iterates that a Random Pursuit algorithm generates on the very simple sphere function $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2$. For iterate $\mathbf{x}_k \in \mathbb{R}^n$, search direction $\mathbf{u}_k \in \mathbb{R}^n$ and a step generated by an exact line search, we have

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathtt{LS}^f(\mathbf{x}_k, \mathbf{u}_k)\mathbf{u}_k = \mathbf{x}_k - \frac{\langle \mathbf{x}_k, \mathbf{u}_k \rangle}{\|\mathbf{u}_k\|^2}\mathbf{u}_k \,. \tag{5.2}$$

This representation follows from the Remark 4.4. We observe that in order to generate the sequence $(\mathbf{x}_k)_{k\geq 0}$, we essentially only need to evaluate the scalar product $\langle \mathbf{x}_k, \mathbf{u}_k \rangle$. Hence, we see that by means of (5.2) we cannot only define a sequence $(\mathbf{x}_k)_{k\geq 0}$ in $\mathbb{R}^n$, but in any space that is equipped with a scalar product. Thus we can analyze (5.2) in general Hilbert spaces $\mathcal{H}$.

Another interpretation of the sequence (5.2) is the following. Let $H_k := \{\mathbf{x} \in \mathcal{H} \colon \langle \mathbf{x}, \mathbf{u}_k \rangle = 0\}$, denote the central hyperplane orthogonal

to $\mathbf{u}_k$. It can easily be verified that $\mathbf{x}_{k+1} \in H_k$, that is, the iterate $\mathbf{x}_{k+1}$ is the projection of $\mathbf{x}_k$ on H_k.

In the following, we discuss two examples: in the first one we consider simply $\mathbb{R}^n$, as we did so far. We can compute the expected convergence of the sequence $(\mathbf{x}_k)_{k\geq 0}$ exactly. As this sequence corresponds to the steps of a Random Pursuit algorithm on the sphere function, the result provides a lower bound on the convergence rate that can be achieved for Random Pursuit algorithms. The second example is in SYM_n, the space of symmetric $n \times n$ matrices.

5.1.1 Random Pursuit on the Reals

We study the squared norm $\|\mathbf{x}\|^2$ of the iterates $\mathbf{x}_k \in \mathbb{R}^n$. By straightforward calculation, we see that

$$\begin{aligned}\|\mathbf{x}_{k+1}\|^2 = \langle \mathbf{x}_{k+1}, \mathbf{x}_{k+1}\rangle &= \|\mathbf{x}_k\|^2 - \frac{\langle \mathbf{x}_k, \mathbf{u}_k\rangle^2}{\|\mathbf{u}_k\|^2} \\ &= \left(1 - \beta_{I_n}(\mathbf{x}_k, \mathbf{u}_k)\right) \|\mathbf{x}_k\|^2 \,.\end{aligned}$$

If the directions $\mathbf{u}_k$ are independent copies of $\mathbf{u} \sim S^{n-1}$, then by Lemma A.7 we have $\mathbb{E}\left[\|\mathbf{x}_{k+1}\|^2 \mid \mathbf{x}_k\right] = \left(1 - \frac{1}{n}\right) \|\mathbf{x}_k\|^2$. For this process, we know the exact (expected) convergence of $\|\mathbf{x}_k\|^2$, and this is also optimal. Let us put this as a remark.

Remark 5.1 (Optimal rate)**.** *Let $f(\mathbf{x}) = \frac{1}{2}\left\|\mathbf{x}^2\right\|$ and $(\mathbf{x}_k)_{k\geq 0}$, $\mathbf{x}_k \in \mathbb{R}^n$ any Local Search Scheme* (3.1) *with isotropic search directions $(\mathbf{u}_k)_{k\geq 0}$, $\mathbf{u}_k \sim S^{n-1}$. Then $\mathbb{E}\left[\|\mathbf{x}_k\|^2\right] \geq \left(1 - \frac{1}{n}\right) \|\mathbf{x}_0\|^2$, for any choice of the step sizes. That is, Random Pursuit* (5.2) *with exact line search and search directions uniformly from S^{n-1} converges with the optimal rate.*

Proof. Due to symmetry of f, it is not hard to argue that the "best" steps are indeed equal to an exact line search. Nonetheless, this lower bound was proven by Jägersküpper [114] in the scope of hit-and-run algorithms. □

5.1.2 Random Pursuit on Symmetric Matrices

Now we consider SYM_n. For $X_k, U_k \in \mathrm{SYM}_n$, equation (5.2) reads as

$$X_{k+1} = X_k - \frac{\langle X_k, U_k\rangle}{\|U_k\|_F^2} U_k \,, \tag{5.3}$$

where $\langle X_k, U_k \rangle = \mathrm{Tr}[X_k^T U_k]$ denotes the standard scalar product in SYM_n.

Remark 5.2 (Properties)**.** *Let $X_k, U_k \in \mathrm{SYM}_n$ and X_{k+1} defined by* (5.3)*. Let $P \in \mathbb{R}^{n\times n}$ orthogonal, i.e. $PP^T = I_n$. Then*

$$\begin{aligned} &(i) \quad \langle X_{k+1}, U_k \rangle = 0 \\ &(ii) \quad \|X_{k+1}\|_F \leq \|X_k\|_F \ , \\ &(iii) \quad PX_{k+1}P^T = PX_kP^T - \left\langle PX_kP^T, PU_kP^T \right\rangle PU_kP^T \ , \end{aligned}$$

i.e. the rotated matrices PX_kP^T and $PX_{k+1}P^T$ satisfy (5.3) *as well.*

Proof. Properties (i) and (ii) follow directly from our observations in the beginning of Section 5.1 and have also been shown by [141]. We prove (iii) on page 121 in the appendix. □

Note that SYM_n can be embedded into $\mathbb{R}^{\frac{n(n+1)}{2}}$. Therefore our observation in the previous section implies the following: Suppose that the search directions are independent copies of $U \sim S^{n^2-1}$, the uniform distribution on the unit sphere, then $\mathbb{E}[\|X_{k+1}\|_F^2 \mid X_k] = \left(1 - \frac{2}{n(n+1)}\right) \|X_k\|_F^2$. The distribution $S_1^{n^2-1}$, that we encountered in Example 4.16 on page 68, can much easier be generated than the uniform distribution, and we now study the convergence if the search directions are copies of $U \sim S_1^{n^2-1}$. This is a direct application of Theorem 3.6.

Corollary 5.3 (Matrix-valued RP)**.** *Let $(X_k)_{k\geq 0}$, $X_k \in \mathrm{SYM}_n$ be a sequence satisfying* (5.3)*, with search directions $(U_k)_{k\geq 0}$, $U_k \in S_1^{n^2-1}$, then*

$$(i) \quad \|X_k\|_F^2 \leq \|X_0\|_F^2 \cdot \exp\left[-\sum_{i=0}^{k-1} \frac{\langle X_i, U_i \rangle^2}{\|X_i\|_F^2} \right] .$$

If U_k are independen copies of $U \sim S_1^{n-1}$. Then

$$\begin{aligned} &(ii) \quad \mathbb{E}\left[\|X_{k+1}\|_F^2 \mid (X_i)_{i=0}^k \right] = \|X_k\|_F^2 - \frac{2\|X_k\|_F^2 + \mathrm{Tr}[X_k]^2}{n(n+2)} \ , \\ &(iii) \quad \mathbb{E}\left[\|X_k\|_F^2 \right] \leq \left(1 - \frac{2}{n(n+2)} \right)^k \|X_0\|_F^2 \ . \end{aligned}$$

Proof. Property (i) is the statement of Theorem 3.3. The one-step progress in (ii) has been calculated in Example 4.16, and (iii) follows therefore from Theorem 3.6. Part (iii) was first shown by [141]. □

The one step progress given in part (ii) of the above statement can be bounded as follows:

$$\frac{2}{n(n+2)} \|X_k\|_F^2 \leq \frac{2\|X_k\|_F^2 + \mathrm{Tr}[X_k]^2}{n(n+2)} \leq \frac{1}{n} \|X_k\|_F^2 , \tag{5.4}$$

where the first inequality is trivial and the second follows by Cauchy-Schwarz: $\mathrm{Tr}[X_k]^2 \leq n \|X_k\|_F^2$. Both the upper and lower bound are tight in general but they differ by a factor of approximately n. Thus, one might wonder if the bound of Leventhal and Levis [141], i.e. (iii) of the above statement, describes the expected behavior tightly, or if a much better bound could be derived. This is very similar to the discussion of the convergence factors in Section 4.3.1 above. We answer this question in the next theorem, that was part of the technical report [238]. It shows, that after a short tune-in phase, the expected norm $\mathbb{E}\big[\|X_k\|_F^2\big]$ converges indeed with the rate $\big(1 - \frac{2}{n(n+2)}\big)$. In the tune-in phase convergence might be faster, but the accumulated effect is very limited: the initial error might be decreased at most by a factor of 2, that is $\mathbb{E}\big[\|X_k\|_F^2\big] \geq \frac{1}{2} \|X_0\|_F^2 \big(1 - \frac{2}{n(n+2)}\big)^k$. We explain this below.

Theorem 5.4 (Exact Matrix valued RP)**.** *Let $(X_k)_{k\geq 0}$ as in Cor. 5.3 above and parameters $\xi_1(k) := \big(\lambda_1^k + \lambda_2^k\big)$, $\xi_2(k) := \big(\lambda_1^k - \lambda_2^k\big)$ 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

$$\mathbb{E}\left[\|X_k\|_F^2\right] = \xi_1(k) \frac{\|X_0\|_F^2}{2} - \xi_2(k) \left(\frac{(2n+1)\|X_0\|_F^2}{2\omega} - \frac{\mathrm{Tr}[X_0]^2}{\omega} \right) ,$$

$$\mathbb{E}\left[\mathrm{Tr}[X_k]^2\right] = \xi_1(k) \frac{\mathrm{Tr}[X_0]^2}{2} - \xi_2(k) \left(\frac{2\|X_0\|_F^2}{\omega} - \frac{(2n+1)\mathrm{Tr}[X_0]^2}{2\omega} \right) .$$

Before sketching the proof of this theorem, let us discuss its statement. Lemma B.4 from the appendix shows that for $n \geq 2$, $\lambda_1 = 1 - \Theta\big(\frac{1}{n}\big)$ and $\lambda_2 = 1 - \Theta\big(\frac{1}{n^2}\big) \leq 1 - \frac{2}{n(n+2)}$. Therefore, $\xi_1(k) \approx$

$-\xi_2(k) \approx \lambda_2^k$ for $(k \to \infty)$. From Cor. 5.3 we know that $\mathbb{E}[\|X_k\|_F^2] \leq \big(1 - \frac{2}{n(n+2)}\big)^k \|X_0\|_F^2$. The exact expression (i) reaches this bound (approximately) if $\mathrm{Tr}[X_0] = 0$. However, if $|\mathrm{Tr}[X_0]|$ is large, $\mathrm{Tr}[X_0]^2 = n\|X_0\|_F^2$, say, then term in the right bracket almost vanishes and $\mathbb{E}[\|X_k\|_F^2] \approx \frac{1}{2}\lambda_2^k \|X_0\|_F^2$. That is, the upper bound from Leventhal and Lewis [141], i.e. part (iii) of Cor. 5.3, is tight: the convergence factor cannot be significantly improved, $\lambda_2 \approx \big(1 - \frac{2}{n(n+2)}\big)$, especially for n large. Thus essentially, $\frac{1}{2}\|X_0\|_F^2 \big(1 - \frac{2}{n(n+2)}\big)^k \lesssim \mathbb{E}\big[\|X_k\|_F^2\big] \lesssim \|X_0\|_F^2 \big(1 - \frac{2}{n(n+2)}\big)^k$.

Proof of Theorem 5.4. By Corollary 5.3 (ii) we have an exact expression for $\mathbb{E}\big[\|X_{k+1}\|_F^2 \mid (X_i)_{i=0}^k\big]$. From equation (5.3) we deduce $\mathrm{Tr}[X_{k+1}] = \mathrm{Tr}[X_k] - \langle X_k, U_k\rangle$. (Note that for $U_k := \mathbf{u}_k\mathbf{u}_k^T$ with $\|\mathbf{u}_k\| = 1$, we have $\mathrm{Tr}[U_k] = \mathrm{Tr}[\mathbf{u}_k\mathbf{u}_k^T] = 1$.) Therefore with Lemma A.7 we obtain

$$\begin{aligned}\mathbb{E}\big[\mathrm{Tr}[X_{k+1}]^2 \mid (X_i)_{i=0}^k\big] &= \mathrm{Tr}[X_k]^2 \\ &\quad - \mathbb{E}\left[2\langle X_k, U_k\rangle \mathrm{Tr}[X_k] - \langle X_k, U_k\rangle^2 \mid X_k\right] \\ &= \left(1 - \frac{2n+3}{n(n+2)}\right)\mathrm{Tr}[X_k]^2 + \frac{2}{n(n+2)}\|X_k\|_F^2 \,.\end{aligned}$$

We obtain a linear recurrence, depending only on $\|X_k\|_F^2$ and $\mathrm{Tr}[X_k]^2$. 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, $\mathbb{E}[E[\|X_{k+1}\|_F^2 \mid (X_i)_{i=0}^k] \mid (X_i)_{i=0}^{k-1}] = \mathbb{E}[\|X_{k+1}\|_F^2 \mid (X_i)_{i=0}^{k-1}]$. Repeating this procedure for $(X_i)_{i=0}^{k-2}$ up to (X_0), we finally obtain $E[\|X_{k+1}\|_F^2 \mid X_0] = \mathbb{E}[\|X_{k+1}\|_F^2]$. We observe that all intermediate expressions only depend linearly on $\|X_0\|_F^2$ and $\mathrm{Tr}[X_0]^2$, that is we can write $\big(\mathbb{E}[\|X_k\|_F^2, \mathbb{E}[\mathrm{Tr}[X_k]^2]\big)^T = C(n)^k\big(\|X_0\|_F^2, \mathrm{Tr}[X_0]^2\big)^T$ 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 B.3 in the appendix. □

5.2 Learning the Hessian

We study the problem of estimating the Hessian matrix of a quadratic function $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_A^2$ with $A \in \mathrm{PD}_n$, given only oracle access to f. In principle, we could do this with the following scheme.

Example 5.5. *Let $\mathbf{e}_i$ for $i = 1, \ldots, n$, denote the standard unit vector in $\mathbb{R}^n$ and let $f(\mathbf{x}) := \frac{1}{2}\|\mathbf{x}\|_A^2$. For $\mathbf{y}, \mathbf{u} \in \mathbb{R}$ we have*

$$f(\mathbf{y} + \mathbf{u}) - 2f(\mathbf{y}) + f(\mathbf{y} - \mathbf{u}) = \mathbf{u}^T A \mathbf{u}\,, \tag{5.5}$$

Therefore, by evaluating f at $\mathbf{y}$ and $\mathbf{x} \pm \mathbf{e}_i$ for $i = 1, \ldots, n$, we can find the diagonal entries A_{ii} for $i = 1, \ldots, n$. By querying the additional points $\mathbf{x} \pm (\mathbf{e}_i + \mathbf{e}_j)$, we find also the off diagonal entries A_{ij} for $1 \leq i, j \leq n$ by observing $(\mathbf{e}_i + \mathbf{e}_j)^T A(\mathbf{e}_i + \mathbf{e}_j) = A_{ii} + 2A_{ij} + A_{jj}$. Thus with a total of $n(n+1)+1$ function evaluations, we find all the entries of $A \in \mathrm{PD}_n$.

This scheme can be applied if the objective function is quadratic, for general convex functions one could approximate the curvature by considering the directions $\epsilon \cdot \mathbf{e}_i$, for small $\epsilon > 0$ instead. All the function evaluations have to be spend in a close neighborhood of $\mathbf{x}$, as the scheme crucially depends on the values on the diagonal. We are looking for a scheme that allows combination of the Hessian estimation process with the search. That is, we would like to gradually improve the Hessian estimation in parallel to the search. This allows to gradually incorporate changes in the curvature, which can be dramatic for non-convex functions. Such a scheme would also be much closer to the Covariance Estimation scheme that is implemented in CMA-ES.

Leventhal and Lewis [141] proposed the Randomized Hessian Estimation (RHE) scheme that attracted our attention because it turns out to be identical to the Random Pursuit algorithm (5.3). The (RHE) generates a sequence $(B_k)_{k\geq 0}$ of matrices, $B_k \in \mathrm{SYM}_n$, that converge: $B_k \to A$ for $k \to \infty$. The scheme takes the form

$$B_{k+1} = B_k + \langle A - B_k, U_k \rangle\, U_k\,, \tag{RHE}$$

for a sequence $(U_k)_{k\geq 0}$ of random rank-one matrices $U_k \sim S_1^{n-1}$. This update requires the value the scalar product $\langle A, U_k \rangle$, where A is the (unknown) metric and cannot be accessed directly. However, as in Example 5.5, the scalar product can be evaluated at the expense of two additional function evaluations because of the simple structure of the directions U_k. Let $\epsilon > 0$, then for $U_k = \mathbf{u}_k \mathbf{u}_k^T$ and $\mathbf{x} \in \mathbb{R}^n$,

$$\langle A, U_k \rangle = \mathbf{u}_k^T A \mathbf{u}_k = \frac{f(\mathbf{x} + \epsilon \mathbf{u}_k) - 2f(\mathbf{x}) + f(\mathbf{x} - \epsilon \mathbf{u}_k)}{\epsilon^2}\,, \tag{5.6}$$

By rewriting the process RHE in terms of the error matrices $E_k := B_k - A$, for all $k \geq 0$, we find $E_{k+1} = E_k - \langle E_k, U_k \rangle\, U_k$, and thus exactly process (5.3)—a Random Pursuit algorithm with exact line search

and search directions sampled from $S_1^{n^2-1}$. As a consequence, our convergence results, especially Corollary 5.3 and Theorem 5.4 do hold for the process (RHE). We here only state the implications on the convergence factor ϱ_A.

Lemma 5.6 (Convergence factor)**.** *Let $A \in \mathrm{PD}_n$ and $(B_k)_{k\geq 0}$ a sequence of matrices $B_k \in \mathrm{SYM}_n$ satisfying* (RHE) *for search directions $(U_k)_{k\geq 0}$, independent copies $U_k \sim S_1^{n^2-1}$. For parameter $b > 1$, define the threshold $K = \lceil n(n+2)\left(\ln \|B_0 - A\|_F + \ln \left\|A^{-1}\right\|_2 + \frac{1}{2}\ln b\right) + 1\rceil$. Then with probability at least $\left(1 - \frac{1}{b}\right)$, $\lambda_{\min}(AB_K^{-1}) > 0$ and for every $j \geq 0$, with probability at least $\left(1 - \frac{1}{b}\right)$:*

$$\varrho_A(B_{K+j}) \leq 1 - \frac{1-\eta^j}{(1+\eta^j)n}\,,$$

where $\eta = \left(1 - \frac{2}{n(n+2)}\right)$.

The proof can be found on page 123 of the appendix.

Remark 5.7. *Note that for $B_0 = 0$, the threshold K only depends on $\|A\|_F \left\|A^{-1}\right\|_2$, the relative condition number of A; denoted as $\kappa_{\mathrm{F}}(A)$ in Example 4.15.*

5.2.1 On the Complexity of Hessian Learning

In the section above, we have seen that (RHE) can be used to estimate the Hessian matrix of a quadratic function using only function values. We now discuss the complexity of this approach

Let $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_A^2$ with $A \in \mathrm{PD}_n$ quadratic as in the previous section. Let $(\mathbf{x}_k)_{k>0}$ with $\|\mathbf{x}_0\| = 1$ denote the iterates of a Random Pursuit algorithm with search directions $(\mathbf{u}_k)_{k\geq 0}$ and steps generated by an exact line search oracle. For parameter $\epsilon > 0$, we denote by $N(\epsilon)$ the complexity as defined in Section 2.2, that is the smallest integer such that $\mathbb{E}[f(\mathbf{x}_{N(\epsilon)})] \leq \epsilon$. If the steps are independent copies of $\mathbf{u} \sim S^{n-1}$, then by Theorem 3.6 and Example 4.9, we have $N(\epsilon) = O\big(n\kappa(A)\ln\frac{1}{\epsilon}\big)$.

On the other hand, we could first estimate an approximation B of the Hessian by (RHE), and then sample the search directions $\mathbf{u} \sim S_B^{n-1}$. By Lemma 5.6 and Remark 5.7, we have for this approach $N(\epsilon) = O\big(n^2 \ln \kappa_{\mathrm{F}}(A) + n\ln\frac{1}{\epsilon}\big)$. In contrast, the running time of a simple Random Pursuit algorithm that does not update the sampling distribution, i.e. $\mathbf{u}_k \sim S^{n-1}$ for every iteration k, is $N(\epsilon) =$

$O\big(n\kappa_{\mathrm{T}}(A)\ln\frac{1}{\epsilon}\big)$. By comparing those expressions, we conclude that Hessian estimation with the scheme (RHE) makes only sense in low dimension $\big(n\ln\kappa_{\mathrm{F}}(A)\leq\kappa(A)\ln\frac{1}{\epsilon}\big)$. In Section 5.2.2 below, we will show show how the dependency of (RHE) on the initial error, i.e. the relative condition number $\kappa_{\mathrm{F}}(A)$, can be avoided. This modified procedure attains $N(\epsilon)=O\big(n^2+n\ln\frac{1}{\epsilon}\big)$—independent of A. Hence, it is affine invariant. However, also this modified scheme is only superior to a simple Random Pursuit if the dimension is not too large, i.e. $n\leq\kappa_{\mathrm{T}}(A)\ln\frac{1}{\epsilon}$.

We would like to emphasize that Hessian estimation is not *per se* a bad approach in high dimensions, but the quadratic dependency on the dimension of the running time of (RHE) does rule out this specific scheme for this task. Hessian estimation can be a crucial ingredient to expedite the convergence of any algorithm. However, schemes for high-dimensional applications applications must find a good approximation B in linear time. This implies that such an approximation must have a linear representation, for instance of the form $B=R+S$, where R is a low rank matrix and S a sparse matrix with only $O(n)$ elements. It is not clear, whether the presented (RHE) can be modified such as to incorporate constraints of this form. Especially, as the modified procedure should converge in linear time to a suitable approximation.

5.2.2 Affine Invariant Hessian Estimation

Let us look at (RHE) from a slightly different angle. The entries of A are $\Theta(n^2)$ unknown variables that must be determined. Each linear measurement as in (5.6) defines one linear equation that is satisfied by the entries of A. Therefore, $\Theta(n^2)$ measurements suffice (see Example 5.5 above), to determine the entries of A completely. Solving this system of linear equations can be done offline, that is, it does not require additional function evaluations, resulting at complexity (in terms of function evaluations) of only $\Theta(n^2)$, independent of A. The linear system could in principle be solved by an arbitrary algorithm (see e.g. Section 5.3 below). However, one of the most elegant choices is to just use (RHE) in an offline fashion.

Example 5.8 (Offline (RHE))**.** *Let $\epsilon>0$ and let $T=(\mathbf{v}_i)_{k=1}^N$ denote a set of $N\geq C(\epsilon)n^2$ independent copies of random directions $\mathbf{v}\sim S^{n-1}$. If $C(\epsilon)$ is chosen as $C(\epsilon,t)$ from Example 4.19 (for arbitrary parameter $t>1$), then with probability at least $1-e^{-t\sqrt{n}}$ the sequence $(B_k)_{k\geq0}$*

generated by (RHE) *with search directions* $\mathbf{u} \sim T$ *converges and satisfies*

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

Proof. This is an immediate application of Theorem 3.6 and Example 4.19. □

The important observation here is that the expensive linear measurements $\langle A, \mathbf{v}_i \mathbf{v}_i^T \rangle$ need only be done once, and then (RHE) can be run offline until the error $\|B_k - A\|_F^2$ is as small as desired. However, note that by just using a fixed set of directions, the scheme looses its ability to adapt to changes in the curvature (if applied to non-quadratic functions). In Section 6.2 we present a concise implementation that uses the idea of Example 5.8, but repeatedly updates the set T of directions.

5.2.3 A Note on General Convex Functions

So far, we have only discussed (RHE) for quadratic functions. In this section, we make some remarks on the general case.

If the objective function is not quadratic, then the crucial equality (5.6) does not hold in general, only in the limit ($\epsilon \to 0$). Leventhal and Lewis [141] therefore suggest to use a small value for ϵ in this case, but no precise error bounds were given. If the measurements (5.6) of the Hessian $\nabla^2 f(\mathbf{x})$ at some $\mathbf{x} \in \mathbb{R}^n$ are not consistent (due to errors), then the corrupted linear system has no solution. However, by the error bounds in Theorem 3.6, we know that in this case the estimations $(B_k)_{k\geq 0}$ generated by (RHE) still approach $\nabla^2 f(\mathbf{x})$, i.e. $\lim_{k\to\infty} \left\|B_k - \nabla^2 f(\mathbf{x})\right\| \leq \delta(\epsilon)$ for some (unspecified) error term $\delta(\epsilon)$ that accounts for the inaccuracy in the measurements (see also [141, Thm. 2.3]). This is one of the advantages, why one would want to use scheme (RHE) at all.

5.2.4 Example and Implementations

We conclude our discussion of the (RHE) scheme by pointing out a few interesting empirical observations. In [235] we tested an implementation of the (RHE) scheme on several objective functions with different spectra. The eigenvalues of the functions have been parameterized to interpolate between the two extreme cases of the spectrum given by the

functions f_{two} and f_{flat} from Table 2.3 on page 38. The Hessian matrices in both functions have the same maximal (L) and minimal (1) eigenvalues. The function f_{two} has two different scales that are distributed evenly among the dimensions. The second function f_{flat} has also two scales, but only one dimension belongs to the small eigenvalue, and $n-1$ to much larger ones (if $L \gg 1$).

We compared the performance of several different zeroth-order algorithms that internally compute an approximation of the Hessian (or its inverse) in [235]. The schemes included a variant of CMA-ES, Gaussian Adaption, and (RHE). One of our observations was that optimization on f_{two} seemed to be slightly more difficult than on f_{flat}. This effect was most pronounced in CMA-ES and Gaussian Adaptation. In a related investigation in [237], we found that certain implementations of (RHE) can be more efficient on f_{flat} than on f_{two}. From a theoretical point of view, the scheme (RHE) should converge with the same rate on both functions (see e.g. Theorem 5.4). We have to conclude that the observed positive effect on f_{flat} depends on the implementation of (RHE). We will now shortly discuss this. To support the discussion, we have depicted the approximation error $\|B_k - A\|_F^2$, for $B_0 = L \cdot I_n$, together with the convergence factor $\varrho_A(B_k)$ in Figure 5.1 for the two functions f_{flat} and f_{two}, with parameter $L = 1\text{E}10$ in $n = 20$ dimensions.

Unconstrained

A straightforward implementation of (RHE) looks as follows: in each iteration k, sample a search direction $U_k \sim S_1^{n^2-1}$ and estimate the curvature according to (5.6). This requires two (additional) function evaluations in each iteration. By the definition of (RHE), all iterates $B_k \in \text{SYM}_n$, however, some of the B_k's might be not be positive definite. Lemma 5.6 shows, that $B_k \in \text{PD}_n$ for k large enough and the convergence factor $\varrho_A(B_k)$ approaches the optimal value $\left(1 - \frac{1}{n}\right)$. This is depicted in Figure 5.1 (where we set $\varrho_A(B_k) = 1$ as long as $B_k \notin \text{PD}_n$).

At the beginning of Section 5.2 we argued that we would like to interlace the search (optimization) process, and the Hessian learning process. In order to sample search directions $\mathbf{u} \sim S_{B_k}^{n-1}$ for an estimate B_k, it is required that $B_k \in \text{PD}_n$. This can be assured by different techniques: (i) project B_k back to PD_n if it falls outside, (ii) simply reject B_k if it falls outside of PD_n, i.e. do not use it for sampling, and (iii) wait until the unconstrained scheme converges. The last idea sounds very costly. However, by storing $\Theta(n^2)$ curvature estimations, we can

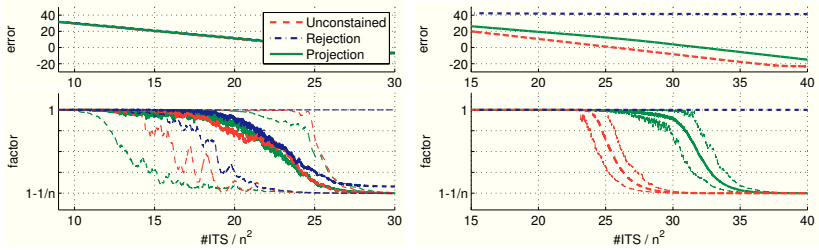

Figure 5.1: Evolution of $\log \|B_k - A\|_F^2$ vs. iterations (#ITS) (top) and $\varrho_A(B_k)$ vs. #ITS (bottom) on f_{flat} (left) and f_{two} (rigth) with $L = 1\text{E}10$ in $n = 20$ dimensions for the three schemes. Mean of 51 runs, max and min attained by $\varrho_A(B_k)$ are indicated by thin lines (smoothed in the left plot).

apply the offline scheme from Example 5.8 on this whole batch and with high probability the set T is diverse enough such that (RHE) converges to a matrix in PD_n that we can use for sampling. A concrete implementation of this is presented later in Section 6.2, denoted as **Lev-Unc**. We now discuss the other two approaches.

Rejection Sampling

The idea of Rejection Sampling is to simply check if a proposed B_{k+1} is positive definite definite and only accept the iterate if $B_{k+1} \in \text{PD}_n$. If the iterate is not accepted, then a new update direction U_k' is sampled and the step is repeated. To check if $B_{k+1} \in \text{PD}_n$, we can use Wedderburn's formula [250, pg. 69]: $A \in \text{PD}_n$, $\mathbf{u} \in S^{n-1}$, the matrix $A + t\mathbf{u}\mathbf{u}^T$ is positive definite if and only if $t^{-1} < \mathbf{u}^T A^{-1}\mathbf{u}$. Therefore $B_{k+1} \in \text{PD}_n$ only if $B_k \in \text{PD}_n$ and

$$(\mathbf{u}_k^T E_k \mathbf{u}_k)(\mathbf{u}_k^T B_k^{-1} \mathbf{u}_k) < 1\,, \tag{5.7}$$

for $E_k := B_k - A$.

This technique is not only expensive (we might need to wait a long time until we are successfully) but we do also not make any progress while waiting. It appears, that the constraint (5.7) has less impact on f_{flat} than on f_{two} (see Fig 5.1). But for both functions the acceptance rate drops dramatically for increasing L, rendering this scheme useless.

Projection Step

By using a projection, we can project any infeasible $B_k \notin \mathrm{PD}_n$ immediately back to the convex set PD_n. If $B_k \in \mathrm{PD}_n$, then, by the fact that the update (RHE) only performs a rank-one perturbation of B_k, Weyl's Theorem implies that at most one eigenvalue of B_{k+1} could be non-positive (see e.g. [107, Theorem 4.3.4]). In Lemma A.8 we show that this defect can be corrected by performing an additional update in the direction of the eigenvector $\mathbf{z}_1 \in S^{n-1}$ corresponding to the smallest eigenvalue of B_{k+1}, i.e. setting $U_{k+1} := \mathbf{z}_1\mathbf{z}_1^T$.

In summary, the concrete iteration of the scheme looks at follows: if $B_k \in \mathrm{PD}_n$, sample $U_k \sim S_1^{n^2-1}$ and estimate the curvature according to (5.6). If $B_k \notin \mathrm{PD}_n$, compute the eigenvector $\mathbf{z}_1$ corresponding to the smallest eigenvalue of B_k, and perform the update along direction $\mathbf{z}_1\mathbf{z}_1^T$. The computation of the eigenvector could for instance be done with Lanczos' method [47, 137, 139]. A concrete implementation of this is presented later in Section 6.2, denoted as **Lev-Proj**.

We cannot prove any strictly positive performance guarantee if the update is applied to direction $\mathbf{z}_1\mathbf{z}_1^T$ instead of a random $U_k \sim S_1^{n^2-1}$, but we know from property (ii) of Remark 5.2 that the Frobenius norm of the error does not increase. In Figure 5.1 we see that apparently the projection steps have no negative impact on the convergence of (RHE) on f_{flat}, but on f_{two} the projection steps are not very effective. We also see that the projections have a positive effect on the convergence factor $\varrho_2(B_k)$ on f_{flat}: the factor can be very close to $\left(1 - \frac{1}{n}\right)$ for some very small values of k, much better than predicted by Lemma B.4. On f_{two} the effect is opposite. The projection steps do not have any (positive) impact on the convergence factor $\varrho_1(B_k)$ and the scheme with projection is inferior to the simple unconstrained scheme. On this function it is not advisable to enforce $B_k \in \mathrm{PD}_n$, in contrast to f_{flat}.

5.3 Kaczmarz' Method

We study a consistent (overdetermined) system of linear equations

$$A\mathbf{x} = \mathbf{b} \qquad \text{for } \mathbf{x} \in \mathbb{R}^n\,, \tag{LIN}$$

for $A \in \mathbb{R}^{m\times n}$ with full rank n, $m \geq n$ and $\mathbf{b} \in \mathbb{R}^m$. This system can be solved with Gaussian Elimination in $O(mn^2)$ time, see e.g. [56, 173, 240]. We will now review Kaczmarz' algoritm [122]—an iterative scheme that *can* have substantially better running time, i.e. independent of m.

Suppose $\mathbf{x}^* \in \mathbb{R}^n$ is a solution to (LIN). We observe that $\mathbf{x}^*$ has to satisfy $\langle \mathbf{a}_i, \mathbf{x}^* \rangle = \mathbf{b}_i$ for $i = 1, \ldots, m$, where $\mathbf{a}_i \in \mathbb{R}^n$ denote the rows of A. In other words, $\mathbf{x}^*$ is contained in the intersection of the hyperplanes $H_i := \{\mathbf{x} \colon \langle \mathbf{a}_i, \mathbf{x} \rangle = \mathbf{b}_i\}$ for $i = 1, \ldots, m$. Kaczmarz' algorithm generates a sequence $(\mathbf{x}_k)_{k \geq 0}$ of approximations that converge to $\mathbf{x}^*$. In each iteration of the scheme, the current iterate $\mathbf{x}_k$ is projected onto *one* of the hyperplanes H_i for $i = 1, \ldots, m$. If hyperplane H_i is picked in iteration k, say, then the projection can be written as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{\mathbf{b}_i - \langle \mathbf{a}_i, \mathbf{x}_k \rangle}{\|\mathbf{a}_i\|_2^2} \mathbf{a}_i \,. \tag{5.8}$$

From this equation we already see the analogy to (5.2). We will show below, that Kaczmarz' algorithm is actually identical to a local search scheme with exact line search (inexact line search is termed over- or under-relaxation in this setting [215]).

Kaczmarz' algorithm [122], is also known as *Algebraic Reconstruction Technique* (ART) in computer tomography [83, 101, 108, 171] or Projection onto Convex Sets (POCS) in signal processing [58, 223]. Many schemes to select the hyperplanes exist. If the hyperplanes are just selected in their order, from 1 up to m, then clearly, the running time cannot be independent of m. It was observed in many applications, see e.g. [56, 165, 174, 240, 242, 251, 265] and references therein, that selecting the hyperplanes at random can significantly speed up the convergence. However, the analysis of such schemes was often not very enlightening, either because it was difficult to compare the rates with other iterative schemes or they contained quantities that were hard to compute [51, 65]. Strohmer and Vershynin [240] analyzed the scheme when the hyperplanes are selected with probability proportional to the squared norm of their normal vectors. Such sampling was previously also proposed in a different context in [64] and yields running time independent of m. Lately, different lines of research produced improvements, or accelerations of this simple algorithm [56, 140, 174]. The scheme can especially also be applied in a noisy setting, see e.g. [172].

Theorem 5.9 (Strohmer and Vershynin [240])**.** *Let $(i_k)_{k \geq 0}$ be a sequence of indices, $i_k \in [m]$, where index $i_k = j$ with probability proportional to $\|\mathbf{a}_j\|_2^2$, and let $(\mathbf{x}_k)_{k \geq 0}$ be a sequence of iterates satisfying* (5.8)*. Then*

$$\mathbb{E}\left[\|\mathbf{x}_k - \mathbf{x}^*\|_2^2\right] \leq \left(1 - \kappa_{\mathrm{F}}^{-2}(A)\right) \|\mathbf{x}_0 - \mathbf{x}^*\|_2^2 \,,$$

where $\kappa_{\mathrm{F}}(A) = \left\|A^{-1}\right\|_2 \cdot \|A\|_F$ *is the* relative condition number *of* A.

Now we will give a proof of this statement. As an alternative to the direct proof in [240], we use the framework developed in Chapters 3 and 4. This amounts to a slightly different presentation, but the essence is still the same. In order to apply the framework, we have to formulate problem (LIN) as a quadratic optimization problem which can be solved by a Random Pursuit algorithm. The representation as a quadratic problem is immediate and appears throughout the literature [140, 173, 240] and we follow [140]. Instead of minimizing $\|\mathbf{x} - \mathbf{x}^*\|$ directly over $\mathbb{R}^n$, we can write $\mathbf{x} = A^T\mathbf{y}$ for $\mathbf{y} \in \mathbb{R}^m$ since A has full rank, and consider the equivalent problem of minimizing $\left\|A^T\mathbf{y} - \mathbf{x}^*\right\|$. Let us define

$$f(\mathbf{y}) := \frac{1}{2}\left\|A^T\mathbf{y}\right\|_2^2 - \left\langle \mathbf{x}^*, A^T\mathbf{y}\right\rangle = \frac{1}{2}\left\|A^T\mathbf{y}\right\|_2^2 - \langle \mathbf{b}, \mathbf{y}\rangle \ .$$

Let $\mathbf{y}_+ = \mathbf{y} + \mathtt{LS}^f(\mathbf{y}, \mathbf{e}_i)\mathbf{e}_i$ denote one step obtained by an exact line search in direction of a standard unit vector $\mathbf{e}_i$. Since f is quadratic, we have an analytic expression for $\mathtt{LS}^f(\mathbf{y}, \mathbf{e}_i)$, given in Remark 4.4, and

$$\mathbf{y}_+ = \mathbf{y} + \frac{\left\langle \mathbf{b} - AA^T\mathbf{y}, \mathbf{e}_i\right\rangle}{\mathbf{e}_i AA^T\mathbf{e}_i}\mathbf{e}_i = \mathbf{y} + \frac{\mathbf{b}_i - \left\langle AA^T\mathbf{y}, \mathbf{e}_i\right\rangle}{\|\mathbf{a}_i\|^2}\mathbf{e}_i \, .$$

To see the equivalence to (5.8), recall that we had performed the transformation $\mathbf{x} = A^T\mathbf{y}$. The corresponding step in $\mathbf{x}$ is thus

$$\mathbf{x}_+ = A^T\mathbf{y} + \frac{\mathbf{b}_i - \left\langle A^T\mathbf{y}, A^T\mathbf{e}_i\right\rangle}{\|\mathbf{a}_i\|^2}A^T\mathbf{e}_i = \mathbf{x} + \frac{b_i - \langle \mathbf{x}, \mathbf{a}_i\rangle}{\|\mathbf{a}_i\|^2}\mathbf{a}_i \, ,$$

exactly what we claimed.

Proof of Fact 5.9. We have outlined above that (5.8) is equivalent to a local search scheme with exact line search on the quadratic function $f(\mathbf{x}) := \frac{1}{2}\|\mathbf{x} - \mathbf{x}^*\|_2^2$. In order to to apply Theorem 3.6 (with $\gamma = 1$), all that is left is a lower bound on $\mathbb{E}[\beta^2]$. By Example 4.15 we have $\mathbb{E}[\beta^2] \geq \left(\left\|A^{-1}\right\|_2 \|A\|_F\right)^{-2} = \kappa_{\mathrm{F}}^{-2}(A)$. □

Chapter 6

Accelerated Search

In this section we review and discuss an acceleration technique that can boost the convergence rate of simple local search schemes. This technique was first developed in 1983 by Nesterov [179] in the context of first-order convex optimization. In order to surpass the convergence rate of Gradient Descent, the Accelerated (or Fast) Gradient method iteratively develops a model of the objective function. In the monograph [182] this mechanism was explained through means of so-called *estimate sequences*. The accelerated method was successfully used to improve the computational complexity of fundamental problems in computer science [16, 17, 28, 42, 183] and in large scale optimization [185]. In [184] it was shown that the acceleration technique cannot only be applied to search schemes with deterministic search directions, but also to certain randomized schemes. This scheme relies on the estimation of directional derivatives by finite-difference approximations. The accelerated (randomized) Coordinate Descent [185] was improved recently by Lee and Sidfort [140], as they present computationally more efficient updates. The latter paper attracted our interest, because the authors present the proof quite elegantly, using *probabilistic estimate sequences*, a generalization of their deterministic counterparts.

The accelerated random schemes [140, 184, 185] internally rely on gradient information, e.g. (approximations) of directional derivatives. In this chapter we investigate the question, if—and to what extent—the information of the line search oracle `LS` from Section 3.4 could be used instead. That is, for our theoretical investigation we assume that the line search oracle is given as a black-box. This research is motivated

by convincing empirical experiments that show convergence of such accelerated methods on a (small) set of benchmark functions. In [239] we empirically tested the accelerated scheme with an exact line search on four convex and one (simple) non-convex function. Recently, in [233] we implemented an instance of the accelerated scheme equipped with the adaptive step size scheme (2.7) that is also used in (1+1)-ES and observed convergence on three quadratic and Rosenbrock's function [212].

This chapter is based on so far unpublished theoretical results and presents ongoing work and preliminary results. To put these results in some perspective, we proceed as follows: in Section 6.1 below, we present and discuss our results. The technical details will be provided later in Sections 6.3 and 6.4. Section 6.3 introduces the concept of estimate sequences—an essential tool for the convergence proof provided in Section 6.4. In Section 6.2 in-between, we present a small numerical comparison of the algorithmic schemes that we have discussed in this thesis: these are Random Pursuit and its accelerated version, and the Hessian Estimation schemes from Section 5.2.

6.1 Summary of the Results

From an abstract point of view, Nesterov's Accelerated Random Gradient method [184] is an iterative scheme that advances simultaneously two sequences $(\mathbf{x}_k)_{k\geq 0}$ and $(\mathbf{y}_k)_{k\geq 0}$ of search points. Each iteration consists of two conceptually different steps: (i) a simple *search step* that, like in the Random Pursuit framework, finds a better search point $\mathbf{x}_{k+1}$ starting from some iterate $\mathbf{y}_k \in \mathbb{R}^n$ (see lines 4–5 in Figure 6.1), and (ii) an advanced *model building* step, that computes a new $\mathbf{y}_{k+1}$ based on the acquired knowledge of the objective function (see lines 6–7 in Figure 6.1). In the Accelerated Random Gradient method both steps are performed with the help of a gradient oracle.

For example, on a quadratic function $f \in C^1_{1,1}(A)$, $A \in \mathrm{PD}_n$, the Accelerated Random Gradient method needs $O\big(\kappa^{1/2}(A) n \ln \frac{1}{\epsilon}\big)$ iterations to find an ϵ-approximate solution (2.1). This is a significant improvement over the running time of a simple Random Pursuit algorithm. For instance, Random Pursuit with search directions sampled $\mathbf{u} \sim S^{n-1}$ and search steps satisfying the sufficient decrease condition (D1) from page 42 with parameters $\gamma = 1$ and $\epsilon = 0$, requires $O\big(\kappa_{\mathrm{T}}(A) n \ln \frac{1}{\epsilon}\big)$ iterations to achieve the same goal, see Example 4.12. This is unfortunate, as in general $\kappa_{\mathrm{T}}(A)$ can be of the same order of magnitude as $\kappa(A)$.

Here, we present two new contributions: first, we show that the search step (i) in the Accelerated Random Gradient method can be replaced by a line search—or any other search step that satisfies the sufficient decrease condition (D1). The running time of this altered scheme is $O\big(\kappa_{\mathrm{T}}^{1/2}(A) n \ln \frac{1}{\epsilon}\big)$ for $f \in C^1_{1,1}(A)$ and $A \in \mathrm{PD}_n$. We detail this result in Corollary 6.4 below. For the case $\kappa_{\mathrm{T}}(A) < \kappa(A)$, this is even an improvement over the results in [184]. Second, we show that if an exact line search is used in the model building step (ii), the resulting scheme needs $O\big(\kappa_{\mathrm{T}}^{1/2}(A)\omega^{1/2}(A) n \ln \frac{1}{\epsilon}\big)$ iterations on quadratic functions $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_A^2$. The parameter $\omega(A)$ is defined in Corollary 6.5 below. Intuitively, $\omega(A)$ is a (relative) upper bound on the length of the expected step with respect to the expected progress. The main issue with this result is that the quantity $\omega(A)$ cannot easily be computed or bounded, that is we do not have a satisfactory upper bound that holds uniformly for all quadratic functions. However, we can at least prove $\omega(A) \leq \kappa_{\mathrm{T}}(A)$, implying that the accelerated scheme converges at least as fast as the simple Random Pursuit schemes. The parameter $\omega(A)$ has to be given as a parameter to the accelerated scheme (see e.g. Figure 6.1) and should be chosen as small as possible as it has a direct impact on the convergence rate.

Our main result is given in Theorem 6.11 on page 103 below. This theorem provides the promised bound on the convergence rate and a concise definition of the algorithmic scheme. There is some freedom how to choose the initial parameters. Specific choices might even have an impact on the numerical stability of the scheme, similar to the discussion in [140]. For the algorithms presented in Figure 6.1, we chose the initial parameters deliberately in such a way that we get a very simple scheme. We refer to this specific instance therefore as Simple Accelerated Random Pursuit (SARP).

In order to state our results formally, we now proceed by providing the definition of a gradient oracle.

6.1.1 Gradient Oracles

Definition 6.1 (Gradient Oracle)**.** *An (unbiased) gradient oracle for a function $f \in C^1$ and sampling distribution π is a function $\mathfrak{g}^f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ with*

$$\mathbb{E}_{\mathbf{u}\sim\pi}\left[\mathfrak{g}^f(\mathbf{x},\mathbf{u})\right] = \nabla f(\mathbf{x})\,, \tag{6.1}$$

for all $\mathbf{x} \in \mathbb{R}^n$. We will abbreviate $\mathfrak{g}^f(\mathbf{x},\mathbf{u}) = \mathfrak{g}^f_{\mathbf{u}}(\mathbf{x})$.

Example 6.2 (Directional Derivative)**.** *Let $f \in C^1$ and $\mathbf{u} \sim S^{n-1}$. Then $\mathfrak{g}^f_{\mathbf{u}}(\mathbf{x}) = n \langle \nabla f(\mathbf{x}), \mathbf{u} \rangle \mathbf{u}$ is a gradient oracle and the second moment satisfies $\mathbb{E}\left[\left\|\mathfrak{g}^f_{\mathbf{u}}(\mathbf{x})\right\|^2\right] = n \left\|\nabla f(\mathbf{x})\right\|^2$.*

Proof. We have $\mathfrak{g}^f_{\mathbf{u}}(\mathbf{x}) = n(\mathbf{u}\mathbf{u}^T)\nabla f(\mathbf{x})$. The first claim follows by linearity of expectation and $\mathbb{E}[\mathbf{u}\mathbf{u}^T] = \frac{1}{n} I_n$ by Lemma A.4 in the appendix on page 113. Analogously for $\left\|\mathfrak{g}^f_{\mathbf{u}}(\mathbf{x})\right\|^2 = n^2 \nabla f(\mathbf{x})^T \mathbf{u}\mathbf{u}^T \nabla f(\mathbf{x})$. □

Example 6.3 (Transformed Exact Line Search)**.** *Let $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_A^2$ for $A \in \mathrm{PD}_n$, $\mathbf{u} \sim S^{n-1}$ and $Q_A = E\left[\frac{\mathbf{u}\mathbf{u}^T}{\|\mathbf{u}\|_A^2}\right]$ (as in Section 4.3.2). Then $\mathfrak{g}^f_{\mathbf{u}}(\mathbf{x}) = -Q_A^{-1} \mathtt{LS}^f(\mathbf{x}, \mathbf{u})\mathbf{u}$ is a gradient oracle.*

Proof. We have $\mathbb{E}\left[\mathtt{LS}^f(\mathbf{x}, \mathbf{u})\mathbf{u}\right] = -Q_A \nabla f(\mathbf{x})$ by definition of the matrix Q_A, see (B.4). □

This last example might seem to have limited practical use: in general we do not know Q_A. Hence, we cannot implement $\mathfrak{g}^f$, we can only access $\mathtt{LS}^f$. For now, we can use it at least use as at theoretical concept, but later in Section 6.4 we show, that in fact we do not need to know Q_A to implement the accelerated search schemes.

6.1.2 Convergence of SARP

As mentioned above, the convergence proof of SARP will be provided in Theorem 6.11 below on page 103. Similar to Theorem 3.3 or 3.6 we show linear convergence

$$\mathbb{E}[f(\mathbf{x}_N)] \leq \psi^k f_0 \,,$$

where $0 < \psi < 1$ denotes the convergence factor and f_N and f_0 are defined as in Theorem 3.3. We now summarize our bounds on ψ.

Corollary 6.4 (SARP with Gradient Oracle)**.** *Let $f \in C^1_{m,L}(A)$ for metric $A \in \mathrm{PD}_n$ and search directions $(\mathbf{u}_k)_{k\geq 0}$ independent copies of $\mathbf{u} \sim S^{n-1}$. For normalization purposes assume without loss of generality $\lambda_{\max}(A) = 1$. If (i) the search steps $\mathbf{x}_{k+1} = \mathbf{y}_k + \sigma_k \mathbf{u}_k$ satisfy the sufficient decrease condition* (D1) *with parameters $\gamma \leq 1$, $\epsilon = 0$ and (ii) the model building steps are generated with gradient oracle $\mathfrak{g}^f_{\mathbf{u}}(\mathbf{x}) = n \langle \nabla f(\mathbf{x}), \mathbf{u} \rangle$, then*

$$\psi \leq \left(1 - \frac{1}{n}\sqrt{\frac{m\gamma}{\kappa_{\mathrm{T}}(A) L}}\right) .$$

SARP$(f, \mathbf{x}_0, N, m, L)$
(with gradient oracle)

$\mathtt{LS}^f$: (γ, ϵ)-line search oracle
$\mathfrak{g}^f$: gradient oracle
$\alpha \leftarrow \frac{1}{n}\sqrt{\frac{\gamma}{L\kappa_{\mathrm{T}}(A)}}$ for $f \in C^1_{m,L}(A)$

1 $\mathbf{y}_0 \leftarrow \mathbf{x}_0$; $\mathbf{v}_0 \leftarrow \mathbf{x}_0$;
2 **for** $k = 1$ *to* N **do**
3 $\quad \mathbf{u}_k \sim S^{n-1}$
4 $\quad \sigma_k \leftarrow \mathtt{LS}^f(\mathbf{y}_{k-1}, \mathbf{u}_k)$
5 $\quad \mathbf{x}_k \leftarrow \mathbf{y}_{k-1} + \sigma_k \mathbf{u}_k$
6 $\quad \mathbf{y}_k \leftarrow (\alpha \mathbf{v}_{k-1} + \mathbf{x}_k)/(1+\alpha)$
7 $\quad \mathbf{v}_k \leftarrow (1-\alpha)\mathbf{v}_{k-1} + \alpha \mathbf{y}_k - \frac{\alpha}{m}\mathfrak{g}^f_{\mathbf{u}_k}(\mathbf{y}_{k-1})$
8 **return** $\mathbf{x}_N$

SARP$(f, \mathbf{x}_0, N, \omega)$
(only with exact line search)

$\mathtt{LS}^f$: exact line search oracle
$\omega(A)$ as small as possible with (6.2)
$\alpha \leftarrow \frac{1}{n}\sqrt{\frac{1}{\omega(A)}}$ for $f \in C^1_{1,1}(A)$

1 $\mathbf{y}_0 \leftarrow \mathbf{x}_0$; $\mathbf{v}_0 \leftarrow \mathbf{x}_0$;
2 **for** $k = 1$ *to* N **do**
3 $\quad \mathbf{u}_k \sim S^{n-1}$
4 $\quad \sigma_k \leftarrow \mathtt{LS}^f(\mathbf{y}_{k-1}, \mathbf{u}_k)$
5 $\quad \mathbf{x}_k \leftarrow \mathbf{y}_{k-1} + \sigma_k \mathbf{u}_k$
6 $\quad \mathbf{y}_k \leftarrow (\alpha \mathbf{v}_{k-1} + \mathbf{x}_k)/(1+\alpha)$
7 $\quad \mathbf{v}_k \leftarrow (1-\alpha)\mathbf{v}_{k-1} + \alpha \mathbf{y}_k + \frac{\alpha n}{\kappa_{\mathrm{T}}(A)}\sigma_k \mathbf{u}_k$
8 **return** $\mathbf{x}_N$

Figure 6.1: Simple Accelerated Random Pursuit (SARP). Fast convergence of the scheme on the left hand side is shown in Cor. 6.4 for $f \in C^1_{m,L}(A)$, convergence for the scheme on the right hand side is shown in Cor. 6.5 for quadratic functions $f \in C^1_{1,1}(A)$, the rate depending on $\omega(A)$ defined in (6.2).

The proof can be found in Section 6.4 on page 105 below. Theorem 6.11 treats also line search oracles with absolute errors $\epsilon > 0$. Considering $\kappa_{\mathrm{T}}(A) \leq \kappa(A)$, we see that the convergence factor can be bounded by $\left(1 - \frac{1}{n}\sqrt{\frac{m\gamma}{\kappa(A)L}}\right)$, instead of only $\left(1 - \frac{1}{n} \cdot \frac{m\gamma}{\kappa(A)L}\right)$ as for the simple Random Pursuit schemes. As a second remark, we would like to mention that if the search steps (i) in the above Corollary 6.4, i.e. lines 4–5 in Figure 6.1, are also generated by the gradient oracle, that is $\mathbf{x}_{k+1} = \mathbf{y}_k + \mathfrak{g}^f_{\mathbf{u}}(\mathbf{y}_k)\mathbf{u}_k$, then the scheme becomes identical to the Accelerated Random Gradient method [184].

Now we investigate what happens when we cannot access directional derivatives but only have a line search oracle.

Corollary 6.5 (SARP with Exact Line Search)**.** *Let $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2_A$ for $A \in \mathrm{PD}_n$, search directions $(\mathbf{u}_k)_{k\geq 0}$ independent copies of $\mathbf{u} \sim S^{n-1}$ and Q_A defined as in Section 4.3.2. If (i) the search steps $\mathbf{x}_{k+1} = \mathbf{y}_k + \mathtt{LS}^f(\mathbf{y}_k, \mathbf{u}_k)\mathbf{u}_k$ for an exact line search oracle, and (ii) the model building steps are generated with gradient oracle $\mathfrak{g}^f_{\mathbf{u}}(\mathbf{x}) = -Q_A^{-1}\mathtt{LS}^f(\mathbf{x}, \mathbf{u})$, then*

$$\psi \leq \left(1 - \frac{1}{n}\sqrt{\frac{1}{\omega(A)\kappa_{\mathrm{T}}(A)}}\right)$$

for parameter $\omega(A)$ (as small as possible) such that

$$\mathbb{E}\left[\left\|\mathtt{LS}^f(\mathbf{x},\mathbf{u})\mathbf{u}\right\|^2_{Q_A^{-1}}\right] \leq \omega(A)\cdot n\cdot \|\nabla f(\mathbf{x})\|^2_{Q_A}\ . \tag{6.2}$$

We further have $\omega(A) \leq \kappa_{\mathrm{T}}(A)$, hence $\psi \leq \left(1-\frac{1}{n\kappa_{\mathrm{T}}(A)}\right)$.

We see, that in case $\omega(A)=1$, the convergence rate can reach the seeked $\left(1-n^{-1}\kappa_{\mathrm{T}}^{-1/2}(A)\right)$. However, in Lemma 4.23 we derived only the upper bound $\omega(A)\leq\kappa_{\mathrm{T}}(A)$. This explains why Lemma 4.23 was termed "insufficient": the bound $\omega(A)\leq\kappa_{\mathrm{T}}(A)$ just allows to prove that SARP with exact line search oracle converges at least as fast as the simple Random Pursuit with exact line search, i.e. with rate $\left(1-n^{-1}\kappa_{\mathrm{T}}^{-1}(A)\right)$; but does not allow for acceleration. We provide the proof of this corollary in Section 6.4.1 on page 106 below.

This version of SARP is depicted in the right panel of Figure 6.1. Besides small additive errors $\epsilon>0$ which are considered in addition in Theorem 6.11, the restriction to an exact line search oracle can be relaxed in two ways: (i) as in the setting of Corollary. 6.4, the search step on lines 4–5 could be replaced with any step that satisfies the sufficient decrease condition (D1) with parameter $\gamma>0$. The the initialization of α must be changed accordingly (as in the left panel of Figure 6.1), and the convergence factor drops in this case to $\psi\leq\left(1-\frac{1}{n}\sqrt{\frac{\gamma}{\omega(A)\kappa_{\mathrm{T}}(A)}}\right)$. However, line 7 still requires an exact line search oracle. Once evaluated, this information could therefore also be used for the update in lines 4–5 and hence this variant might be less useful.

The exact line search oracle in line 7 could also be replaced. The matrix Q_A from Section 4.3.2 was defined such as to describe the expected step $\mathbb{E}[\mathtt{LS}^f(\mathbf{y},\mathbf{u})\mathbf{u}] = -Q_A\nabla f(\mathbf{y})$ of Random Pursuit on a quadratic function f. The here presented technique allows to use inexact line search oracles on line 7 if (ii) $\mathbb{E}[\mathtt{LS}^f(\mathbf{y},\mathbf{u})\mathbf{u}] = -Q'_A\nabla f(\mathbf{y})$ holds for any Q'_A independent of $\mathbf{y}\in\mathbb{R}^n$. This holds for instance for the oracles $\gamma\mathtt{LS}^f(\mathbf{y},\mathbf{u})$, $\gamma>0$, or $\delta\mathtt{LS}^f(\mathbf{y},\mathbf{u})$ where $\delta>0$ is a random variable independent from $\mathbf{x}$. However, to determine the exact convergence rate, bounds analogous to Lemmas 4.22 and 4.23 must be derived for this matrix Q'_A instead. This could also be a way to prove convergence of the accelerated scheme with the adaptive step size control (2.7) (that we implemented in Section 6.2 and in [233])—if such a linear transformation Q'_A exists. This could be the case for quadratic functions.

6.2 Numerical Demonstration

In this section we numerically compare three Random Pursuit algorithms that we encountered in this thesis. These algorithms use different strategies to accelerate the search: (i) none, i.e. we simply implement a plain Random Pursuit algorithm, (ii) variable metric, e.g. the Hessian estimation scheme (RHE) combined with Random Pursuit and (iii) the acceleration technique that we discuss in this chapter.

We implement all algorithmic schemes with the adaptive step size control described in (2.7) (for initial value $\sigma_0 = 1$ and $c = 0.27$). This should emphasize the fact that the schemes do not rely on exact line search oracles. Whilst we have discussed this for the simple schemes already, this is not so clear (from a theoretical point of view) for SARP. Below, we list very briefly the implementation details.

Benchmark Functions

We now detail the test functions used for the numerical comparison. We tested all schemes on three quadratic functions and on one non-convex function, all listed in Table 2.3 on page 38.

The quadratic functions are f_{exp}, f_{lin} and f_{two}, for curvature parameter $L = 1\text{E}6$. The eigenvalues of f_{lin} are equally spaced between 1 and L, the eigenvalues of f_{exp} exponentially, and the eigenvalues of f_{two} take only two different values. The minimum is attained at $\mathbf{x}^* = \mathbf{0}_n$, we start the search at $\mathbf{x}_0 = \mathbf{1}_n$.

The function f_{rosen} is known as Rosenbrock function [212] and non-convex. Yet, it behaves locally almost like a quadratic function. It serves as a test model with smoothly changing Hessian in order to study the valley-following abilities of the different variable-metric schemes. The minimum is attained at $\mathbf{x}^* = \mathbf{1}_n$, we start the search at $\mathbf{x}_0 = \mathbf{0}_n$.

Algorithmic Schemes

RP. We sample the search directions uniformly from $\mathbf{u} \sim S^{n-1}$.
Lev-Proj. This scheme combines the simple Random Pursuit algorithm with the Hessian learning scheme (RHE) from Leventhal and Lewis [141]. We implement the projection step as described in detail in Section 5.2.4. We initialize (RHE) with $B_0 = I_n$. In every iteration k, the objective function is evaluated at exactly three points: two function values are used for the curvature estimation according to (5.6), and an updated Hessian matrix B_k is computed. We use $\epsilon = 1\text{E-}6$

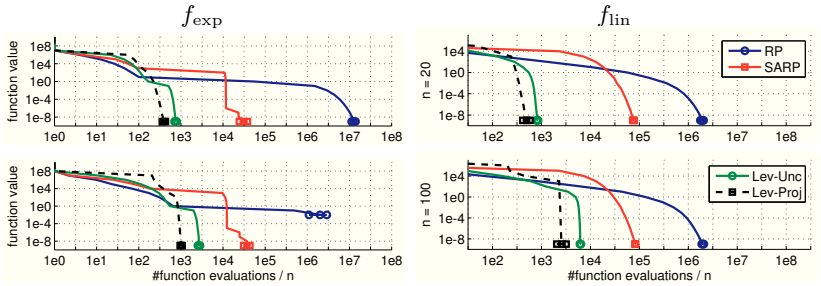

Figure 6.2: Evolution of $\log f$ vs. function evaluations (#FVALS) on f_{exp} (left) and f_{lin} (right) with $L = 1\text{E}6$ in $n = 20$ dimensions (top) and $n = 100$ (bottom). For 51 runs we recorded #FVALS needed to reach function value of 1E-9. The trajectory realizing the median values is depicted, mean and one standard deviation are indicated by markers.

in (5.6) on f_{rosen}. If B_k is positive semidefinite, a search direction $\mathbf{u}_k \sim S^{n-1}_{B_k^{-1}}$ is sampled, otherwise $\mathbf{u}_k \sim S^{n-1}_{B_{k-1}^{-1}}$, as one of the consecutive pair (B_{k-1}, B_k) has to be positive definite. The third function evaluation is then used to evaluate the search step along this direction.

Lev-Unc. This scheme combines the ideas presented in Example 5.8 with the unconstrained scheme from Section 5.2.4. The scheme stores n^2 randomly sampled directions, together with their respective curvature estimates, and processes them in batches. Every batch consists of n^2 iterations; in each iteration three function evaluations take place: two function values are used for the curvature estimate (5.6), and one for the search. Search directions are sampled from $\mathbf{u} \sim S^{n-1}_{B_b^{-1}}$, where B_b remains fixed throughout the whole batch. At the end of each batch b, the stored information is used to run (RHE) in an off-line way: (RHE) subsamples the set of stored directions until the scheme converges to a new estimate B_{b+1}. The subsampling does not require additional function evaluations. In the first batch, the search process is omitted.

SARP. This is the new scheme that we introduce in this chapter. It maintains two sequences $(\mathbf{x}_k)_{k\geq 0}$, $(\mathbf{y}_k)_{k\geq 0}$ of iterates and an auxiliary sequence $(\mathbf{v}_k)_{k\geq 0}$, as detailed in Figure 6.1 (the scheme in the right panel). As we use adaptive step size control, we have to modify the lines 3–5 from the scheme depicted in Figure 6.1. We use the following approach: The index k counts only the successful steps, that is, in iteration k, the adaptive step size (2.7) samples search points near $\mathbf{y}_{k-1}$, until a better point is found, denoted as $\mathbf{x}_k$. Then the sequences get

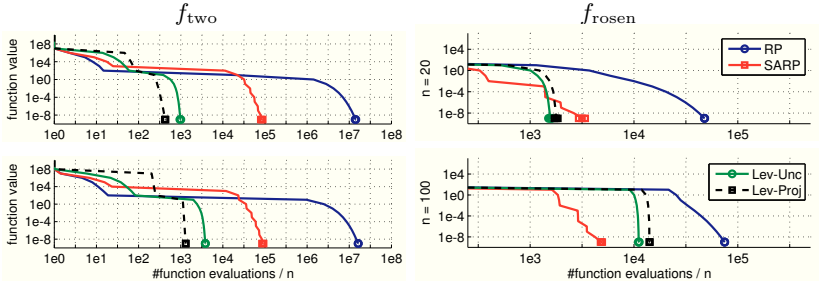

Figure 6.3: Evolution of $\log f$ vs. function evaluations (#FVALS) on f_{two} with $L = 1\text{E}6$ (left) and f_{rosen} (right) in $n = 20$ dimensions (top) and $n = 100$ (bottom). For 51 runs we recorded #FVALS needed to reach function value of 1E-9. The trajectory realizing the median values is depicted, mean and one standard deviation are indicated by markers.

updated as follows: $\mathbf{y}_k$ is computed from formula (6.10); $\mathbf{v}_k$ is computed from formula (6.6), i.e. as shown in lines 6–7 in the right panel of Figure 6.1. These formulas depend on parameters. We initialize $\mathbf{v}_0 = \mathbf{x}_0$, $\mathbf{y}_0 = \mathbf{x}_0$ and use $\zeta_0 = m = 1$, yielding $\zeta_k = 1$ for all k, see (6.5). The parameter ω is, quite arbitrarily from a theoretical point of view, set to $\omega = 1$. For the non-convex f_{rosen} we use the choice $\kappa_{\text{T}} = 500$.

Discussion of the Results

The experimental result are depicted in Figures 6.2 and 6.3. The simple Random Pursuit **RP** needs the most function evaluations on all functions, as expected. The performance, i.e. the number of function evaluations to reach the target accuracy, of the two Hessian estimation schemes **Lev-Proj** and **Lev-Unc** is approximately the same on all functions. However, the first scheme requires expensive eigenvalue computations every few iterations (especially on f_{elli}), whilst the iterations of the latter are computationally less expensive.

The accelerated scheme **SARP** works on all functions consistently better than **RP** and on f_{rosen} it is even faster than the Hessian learning variants. While those schemes need at least n^2 many function evaluations, **RP** and **SARP** scale only linearly in the dimension. Thus, the Hessian learning schemes can only be applied in relatively small dimensions, whist the latter might also be applicable for high dimensional problems. This fact—and the promising empirical performance—motivated our research in this area.

6.3 Estimate Sequence Method

Here we now focus on the essential technical ingredients to prove our claims made in Section 6.1. For this, we extend the method of estimate sequences developed in [184] to the probabilistic setting. First, we introduce this concept formally, and present some fundamental results, mostly adopted from the presentation due to Baes [14].

Definition 6.6 (Estimate Sequence)**.** *A* (probabilistic) estimate sequence *for a convex function* $f\colon \mathbb{R}^n \to \mathbb{R}$ *is a sequence of convex functions* $(\phi_k)_{k\geq 0}$*, each* $\phi_k\colon \mathbb{R}^n \to \mathbb{R}$*, and a sequence of positive numbers* $(\eta_k)_{k\geq 0}$ *satisfying* $\lim_{k\to\infty} \eta_k = 0$ *and*

$$\mathbb{E}[\phi_k(\mathbf{x})] \leq (1-\eta_k) f(\mathbf{x}) + \eta_k \mathbb{E}[\phi_0(\mathbf{x})]\,, \tag{6.3}$$

for all $\mathbf{x} \in \mathbb{R}^n$*,* $k \geq 1$*.*

6.3.1 Facts

The two lemmas below are just direct adaptations from [14]—added for completeness. The first one gives a bound on the convergence rate. From the definition (6.3) we see that an estimate sequence is essentially a sequence of functions whose limit—if it exists—is a lower bound on f. A sequence $(\mathbf{x}_k)_{k\geq 0}$ of points with function values $f(\mathbf{x}_k) \leq \phi_k(\mathbf{x}_k)$ is therefore "trapped" between the estimate sequence $(\phi_k)_{k\geq 0}$ and f, and $(f(\mathbf{x}_k))_{k\geq 0}$ has to converge to $f^* := \min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$. We also see that the convergence rate is fully determined by the sequence $(\eta_k)_{k\geq 0}$.

Lemma 6.7. *Let* $(c_k)_{k\geq 0}$ *be a sequence of positive numbers,* $f\colon \mathbb{R}^n \to \mathbb{R}$ *convex and* $((\phi_k)_{k\geq 0}, (\eta_k)_{k\geq 0})$ *an estimate sequence* (6.3) *for* f*. Suppose that the sequence* $(\mathbf{x}_k)_{k\geq 0}$ *for* $\mathbf{x}_k \in \mathbb{R}^n$*, satisfies* $\mathbb{E}[f(\mathbf{x}_k)] - c_k \leq \min_{\mathbf{x}} \mathbb{E}[\phi_k(\mathbf{x})]$*. Then*

$$\mathbb{E}[f(\mathbf{x}_k)] - f^* \leq \eta_k(\mathbb{E}[\phi_0(\mathbf{x}^*)] - f^*) + c_k\,,$$

for every $k \geq 1$*.*

Proof. It suffices to write

$$\begin{aligned}\mathbb{E}[f(\mathbf{x}_k)] &\leq \min_{\mathbf{x}} \mathbb{E}[\phi_k(\mathbf{x})] + \epsilon_k \leq \min_{\mathbf{x}} f(\mathbf{x}) + \eta_k(\mathbb{E}[\phi_0(\mathbf{x})] - f(\mathbf{x})) + c_k \\ &\leq f^* + \eta_k(\mathbb{E}[\phi_0(\mathbf{x}^*)] - f^*) + c_k\,.\end{aligned}$$

□

A way to construct an estimate sequence, is to recursively build the convex combination of the previous element ϕ_k in the sequence, and an (arbitrary) function $f_k \colon \mathbb{R}^n \to \mathbb{R}$ that is a lower bound on f in expectation (in the probabilistic setting).

Lemma 6.8. *Let $f \colon \mathbb{R}^n \to \mathbb{R}$ convex and and let $\phi_0 \colon \mathbb{R}^n \to \mathbb{R}$ be a convex function such that $\min_{\mathbf{x}} \phi_0(\mathbf{x}) \geq \min_{\mathbf{x}} f(\mathbf{x})$. Let $(\alpha_k)_{k\geq 0}$ with $0 < \alpha_k < 1$ be a sequence whose sum diverges. Suppose also that we have a sequence $(f_k)_{k\geq 0}$, $f_k \colon \mathbb{R}^n \to \mathbb{R}$, of random functions (i.e., each f_k is sampled from a probability distribution π_k) that underestimate f in expectation:*

$$\mathbb{E}[f_k(\mathbf{x})] \leq f(\mathbf{x})$$

for all $\mathbf{x} \in \mathbb{R}^n$ and $k \geq 0$. We define recursively $\eta_0 := 1, \eta_{k+1} := \eta_k(1-\alpha_k)$, and

$$\phi_{k+1}(\mathbf{x}) := (1-\alpha_k)\phi_k(\mathbf{x}) + \alpha_k f_k(\mathbf{x}) = \eta_{k+1}\phi_0(\mathbf{x}) + \sum_{l=0}^{k} \frac{\eta_{k+1}\alpha_l}{\eta_{l+1}} f_l(\mathbf{x}) \,,$$

for all $k \geq 0$. Then $((\phi_k)_{k\geq 0}, (\eta_k)_{k\geq 0})$ is an estimate sequence.

Proof. This is an adaptation of Proposition 2.2 in [14]. Since

$$\ln \eta_{k+1} = \sum_{j=0}^{k} \ln(1-\alpha_j) \leq -\sum_{j=0}^{k} \alpha_j$$

for each $k \geq 0$, the sequence $(\eta_k)_{k\geq 0}$ converges to zero, as the sum of the α_j's diverges. Now let us proceed by induction. Clearly,

$$\begin{aligned}
\mathbb{E}\left[\phi_1(\mathbf{x})\right] &= \mathbb{E}\left[(1-\alpha_0)\phi_0(\mathbf{x}) + \alpha_0 f_0(\mathbf{x})\right] \\
&= \eta_1 \mathbb{E}\left[\phi_0(\mathbf{x})\right] + (1-\eta_1)\mathbb{E}\left[f_0(\mathbf{x})\right] \\
&\leq \eta_1 \mathbb{E}\left[\phi_0(\mathbf{x})\right] + (1-\eta_1) f(\mathbf{x}) \,.
\end{aligned}$$

Now for $k > 1$:

$$\begin{aligned}
\mathbb{E}\left[\phi_{k+1}(\mathbf{x})\right] &= \mathbb{E}\left[(1-\alpha_k)\phi_k(\mathbf{x}) + \alpha_k f_k(\mathbf{x})\right] \\
&\leq (1-\alpha_k)(1-\eta_k) f(\mathbf{x}) + (1-\alpha_k)\eta_k \mathbb{E}\left[\phi_0(\mathbf{x})\right] + \alpha_k f(\mathbf{x}) \\
&= (1-\eta_{k+1}) f(\mathbf{x}) + \eta_{k+1}\mathbb{E}\left[\phi_0(\mathbf{x})\right] \,.
\end{aligned}$$

□

6.3.2 Probabilistic Construction

As an immediate consequence of Lemma 6.8 we see that we can construct an estimate sequence as soon as we have access to lower bounds on f. Suppose $f \in C^1$ and let $\mathfrak{g}^f$, denote an unbiased gradient oracle (6.1) with $\mathbb{E}[\mathfrak{g}^f_{\mathbf{u}}(\mathbf{x})] = \nabla f(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^n$. Then it is not hard to see, that for sequences $(\mathbf{y}_k)_{k\geq 0}$ with $\mathbf{y}_k \in \mathbb{R}^n$ and directions $(\mathbf{u}_k)_{k\geq 0}$, $\mathbf{u}_k \sim S^{n-1}$,

$$\left(f(\mathbf{y}_k) + \left\langle \mathfrak{g}^f_{\mathbf{u}_k}(\mathbf{y}_k), \mathbf{x} - \mathbf{y}_k \right\rangle\right)_{k\geq 0}, \tag{6.4}$$

is a sequence of underestimates in variable $\mathbf{x} \in \mathbb{R}^n$ that satisfy the conditions of Lemma 6.8. In expectation, these linear functions are just the lower bounds (2.9) that exist for every convex function. In the corollary below, we use quadratic lower bounds instead of the linear ones above.

Corollary 6.9 (Gradient Oracle)**.** *Let $f \in C^1_{m,L}(B)$ strongly convex in metric $B \in \mathrm{PD}_n$. Let $\phi_0 \colon \mathbb{R}^n \to \mathbb{R}$ be a convex function such that $\min_{\mathbf{x}\in\mathbb{R}^n} \phi_0(\mathbf{x}) \geq \min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$. Let $(\alpha_k)_{k\geq 0} \in (0,1)$ be a sequence whose sum diverges. Let $(\mathbf{y}_k)_{k\geq 0}$, with $\mathbf{y}_k \in \mathbb{R}^n$ be an arbitrary sequence of candidates and let directions $(\mathbf{u}_k)_{k\geq 0}$ be independent copies of $\mathbf{u} \sim S^{n-1}$. Let $\mathfrak{g}^f$ be an unbiased gradient oracle (6.1) for f. We define recursively $\eta_0 := 1, \eta_{k+1} := \eta_k(1-\alpha_k)$, and*

$$\phi_{k+1}(\mathbf{x}) := (1-\alpha_k)\phi_k(\mathbf{x}) + \alpha_k \Big[f(\mathbf{y}_k) + \left\langle \mathfrak{g}^f_{\mathbf{u}_k}(\mathbf{y}_k), \mathbf{x} - \mathbf{y}_k \right\rangle + \frac{m}{2} \left\| \mathbf{x} - \mathbf{y}_k \right\|_B^2 \Big],$$

for all $k \geq 0$. Then $((\phi_k)_{k\geq 0}, (\eta_k)_{k\geq 0})$ is an estimate sequence for f.

The next lemma is mainly due to Lee and Sidford [140]. We only extended the statement slightly, to accommodate for gradient oracles as defined in (6.1), that is with respect to search directions $\mathbf{u} \sim S^{n-1}$.

Lemma 6.10 (Quadratic Estimate Sequence)**.** *Let the first function be quadratic, $\phi_0 = \Phi_0^* + \frac{\zeta_0}{2} \left\| \mathbf{x} - \mathbf{v}_0 \right\|_B^2$ for $\Phi_0^* \in \mathbb{R}$ and $\zeta_0 > 0$. Then the estimate sequence defined in Cor. 6.9 consists of quadratic functions of*

the form $\phi_k(\mathbf{x}) = \Phi_k^* + \frac{\zeta_k}{2}\|\mathbf{x}-\mathbf{v}_k\|_B^2$, *where*

$$\zeta_{k+1} = (1-\alpha_k)\zeta_k + \alpha_k m\,, \tag{6.5}$$

$$\mathbf{v}_{k+1} = \frac{1}{\zeta_{k+1}}\left[(1-\alpha_k)\zeta_k\mathbf{v}_k + \alpha_k m\mathbf{y}_k - \alpha_k B^{-1}\mathfrak{g}^f_{\mathbf{u}_k}(\mathbf{y}_k)\right]\,, \tag{6.6}$$

$$\begin{aligned}\Phi_{k+1}^* = (1-\alpha_k)\Phi_k^* + \alpha_k f(\mathbf{y}_k) - \frac{\alpha_k^2}{2\zeta_{k+1}}\left\|\mathfrak{g}^f_{\mathbf{u}_k}(\mathbf{y}_k)\right\|^2_{B^{-1}} \\ + \frac{\alpha_k(1-\alpha_k)\zeta_k}{\zeta_{k+1}}\left(\frac{m}{2}\|\mathbf{v}_k-\mathbf{y}_k\|_B^2 + \left\langle\mathfrak{g}^f_{\mathbf{u}_k}(\mathbf{y}_k), \mathbf{v}_k-\mathbf{y}_k\right\rangle\right)\,.\end{aligned} \tag{6.7}$$

Proof. The proof is almost identical to the proof in [140] and thus omitted here. For the sake of completeness, the details can be found in the appendix on page 123. □

6.4 Acceleration with Gradient Oracles

We now show how to use the estimate sequences to design accelerated local search schemes. We assume access to an unbiased gradient oracle for the model building step and (expected) sufficient decrease (E1) from page 47 for the search step.

Theorem 6.11 (Simple Accelerated Random Search (SARP))**.** *Let* $f \in C^1_{m,L}(B)$ *strongly convex in metric* $B \in \mathrm{PD}_n$ *and search directions* $(\mathbf{u}_k)_{k\geq 0}$ *independent copies of* $\mathbf{u} \sim S^{n-1}$. *Let* $\mathfrak{g}^f$ *be an unbiased gradient oracle* (6.1) *for* f. *Let* $(\mathbf{x}_k)_{k\geq 0}$, $\mathbf{x}_k \in \mathbb{R}^n$ *and* $(\mathbf{y}_k)_{k\geq 0}$, $\mathbf{y}_k \in \mathbb{R}^n$, *two sequences of iterates (specified below), each pair* $(\mathbf{y}_k, \mathbf{x}_{k+1})$ *for* $k \geq 0$ *satisfying the sufficient decrease condition*

$$f(\mathbf{y}_k) - \mathbb{E}\left[f(\mathbf{x}_{k+1}) \mid \mathbf{y}_k\right] \geq \frac{\alpha^2}{2}\mathbb{E}\left[\left\|\mathfrak{g}^f_{\mathbf{u}_k}(\mathbf{y}_k)\right\|^2_{B^{-1}}\right] - \epsilon_k\,, \tag{6.8}$$

for a constant $\alpha > 0$ *and sequence of positive errors* $(\epsilon_k)_{k\geq 0}$. *We initialize:*

$$\mathbf{y}_0 = \mathbf{x}_0\,, \qquad \Phi_0^* = f(\mathbf{x}_0)\,, \qquad \phi_0(\mathbf{x}) = \Phi_0^* + \frac{\zeta_0}{2}\|\mathbf{x}-\mathbf{x}_0\|_B^2\,, \tag{6.9}$$

for any $\zeta_0 \geq m$. *Applying Lemma 6.10 with these parameters, choosing* $\alpha_k^2 = \zeta_{k+1}\alpha^2$, *each* $\mathbf{x}_{k+1}$ *satisfying* (6.8) *with respect to* $\mathbf{y}_k$, *and* $\mathbf{y}_k$ *such that:*

$$\frac{\alpha_k\zeta_k}{\zeta_{k+1}}(\mathbf{v}_k-\mathbf{y}_k) + \mathbf{x}_k - \mathbf{y}_k = 0 \tag{6.10}$$

yields an estimate sequence with

$$\mathbb{E}[f(\mathbf{x}_N) - f^*] \leq \psi^N \cdot \left(f(\mathbf{x}_0) - f^* + \frac{m}{2}\left\|\mathbf{x}^* - \mathbf{x}_0\right\|^2\right) + C_N\,,$$

where convergence factor $\psi^k \leq \min\left\{(1-\alpha\sqrt{m})^k, (1+k\alpha\sqrt{m}/2)^{-2}\right\}$ *and* $C_N := \epsilon_N + \sum_{i=1}^{N-1}\prod_{j=N-i+1}^{N}(1-\alpha_j)\epsilon_{N-i}$.

Proof. By construction (and Lemma 6.10) we know that the sequence $((\phi_k)_{k\geq 0}, (\eta_k)_{k\geq 0})$ for $\eta_k = (1-\alpha_k)\eta_{k-1}$ is an estimate sequence. By Lemma 6.7 it remains to show that the sequence $(\mathbf{x}_k)_{k\geq 0}$ stays (almost) always below the minimum of ϕ_k, i.e. satisfies $\mathbb{E}[f(\mathbf{x}_k)] - C_k \leq \min_{\mathbf{x}\in\mathbb{R}^n}\mathbb{E}[\phi_k(\mathbf{x})]$. We prove this claim by induction. The base case $f(\mathbf{x}_0) \leq \phi_0(\mathbf{x})$ follows by choice of Φ_0^*. With Lemma 6.10 and the induction hypothesis, we get

$$\begin{aligned}\mathbb{E}[\Phi_{k+1}^*] \geq \mathbb{E}\Bigg[&(1-\alpha_k)f(\mathbf{x}_k) + \alpha_k f(\mathbf{y}_k) - \frac{\alpha_k^2}{2\zeta_{k+1}}\left\|\mathfrak{g}_{\mathbf{u}_k}^f(\mathbf{y}_k)\right\|_{B^{-1}}^2 \\ &+ \frac{\alpha_k(1-\alpha_k)\zeta_k}{\zeta_{k+1}}\left(\frac{m}{2}\left\|\mathbf{v}_k - \mathbf{y}_k\right\|_B^2 + \left\langle \mathfrak{g}_{\mathbf{u}_k}^f(\mathbf{y}_k), \mathbf{v}_k - \mathbf{y}_k\right\rangle\right)\Bigg] \\ &- (1-\alpha_k)C_k\end{aligned}$$

By convexity $f(\mathbf{x}_k) \geq f(\mathbf{y}_k) + \langle \nabla f(\mathbf{y}_k), \mathbf{x}_k - \mathbf{y}_k\rangle$. Using the property $\mathbb{E}[\mathfrak{g}_{\mathbf{u}_k}^f(\mathbf{y}_k)] = \nabla f(\mathbf{y}_k)$, we obtain:

$$\begin{aligned}\mathbb{E}[\Phi_{k+1}^*] \geq \mathbb{E}\left[f(\mathbf{y}_k) - \frac{\alpha_k^2}{2\zeta_{k+1}}\left\|\mathfrak{g}_{\mathbf{u}_k}(\mathbf{y}_k)\right\|_{B^{-1}}^2\right] - (1-\alpha_k)C_k \\ + (1-\alpha_k)\left\langle \nabla f(\mathbf{y}_k), \frac{\alpha_k\zeta_k}{\zeta_{k+1}}(\mathbf{v}_k - \mathbf{y}_k) + \mathbf{x}_k - \mathbf{y}_k\right\rangle\end{aligned}$$

We see that $\mathbf{y}_k$ was chosen in (6.10) to cancel the third term, so that

$$\mathbb{E}[\Phi_{k+1}^*] \geq f(\mathbf{y}_k) - \frac{\alpha_k^2}{2\zeta_{k+1}}\mathbb{E}\left[\left\|\mathfrak{g}_{\mathbf{u}_k}^f(\mathbf{y}_k)\right\|_{B^{-1}}^2\right] - (1-\alpha_k)C_k\,. \tag{6.11}$$

We have $\alpha_k^2 = \zeta_{k+1}\alpha^2$ so that the induction step follows by assumption (6.8). Finally, we note $\alpha_k^2 \geq m\alpha^2$ for all k, therefore $\prod_{i=0}^{k-1}(1-\alpha_i) \leq (1-\alpha\sqrt{m})^k$. On the other hand, $\psi_{k+1}^{-1/2} \geq 1 + \frac{k\alpha\sqrt{m}}{2}$, see e.g. [182, Lem 2.2.4] or [184]. □

As in Remark 3.4 from Section 3.2 the additive error C_N in Theorem 6.11 can easily be bounded.

Remark 6.12. *Let $\epsilon_k \leq \epsilon$ for $k = 0, \dots, N-1$. Then $C_N \leq \frac{\epsilon}{\alpha\sqrt{m}}$.*

Proof. We observe $\alpha_k^2 \geq m\alpha^2$ for all k, and the claim follows from

$$C_N \leq \epsilon \sum_{i=1}^{N-1} (1 - \alpha\sqrt{m})^i \leq \frac{\epsilon}{\alpha\sqrt{m}}\,. \qquad \square$$

6.4.1 Convergence of Two SARP Instances

Finally, we can prove the convergence of the two SARP instances presented in Figure 6.1. In order to apply Theorem 6.11 from above, we essentially have to determine two different kinds of parameters. These are the parameters m and L for the quadratic lower bound (2.11) and a parameter α that measures the one step progress. All these parameters depend on the metric B, which can be chosen as we please.

Proof of Corollary 6.4. We use Theorem 6.11 with $B = I_n$ and $\zeta_k = m$ for $k \geq 0$. An lower bound for α follows from the sufficient decrease condition (D2) and Example 4.12. Similar as in (E1) on page 47 we have

$$f(\mathbf{y}_k) - \mathbb{E}\left[f(\mathbf{x}_{k+1}) \mid \mathbf{y}_k\right] \geq \frac{\gamma}{2n\kappa_{\mathrm{T}}(A)L} \left\|\nabla f(\mathbf{y}_k)\right\|_{A^{-1}}^2 - \epsilon\,,$$

and on the other hand $\mathbb{E}\left[\left\|\mathfrak{g}_{\mathbf{u}_k}^f(\mathbf{y}_k)\right\|^2\right] = n\left\|\nabla f(\mathbf{y}_k)\right\|^2$ by Example 6.2, and $\left\|\nabla f(\mathbf{y}_k)\right\|^2 \leq \lambda_{\max}(A)\left\|\nabla f(\mathbf{y}_k)\right\|_{A^{-1}}^2 = \left\|\nabla f(\mathbf{y}_k)\right\|_{A^{-1}}^2$ by Lem. 2.2 and the assumption $\lambda_{\max}(A) = 1$. Therefore we can choose α in (6.8) as least as large as to satisfy $\alpha^2 n^2 \kappa_{\mathrm{T}}(A) L = \gamma$, and the claimed bound on $\psi_k \leq (1 - \alpha\sqrt{m})^k$ follows. $\square$

At last, we provide the proof of Corollary 6.5. The use of the gradient oracle $\mathfrak{g}_{\mathbf{u}}^f(\mathbf{x}) = -nQ_A^{-1}\mathtt{LS}^f(\mathbf{x}, \mathbf{u})$ from Example 6.3 is quite delicate. Apparently, to compute $\mathfrak{g}_{\mathbf{u}}^f(\mathbf{x})$ one has to know Q_A—and we don't do that in general. However, there is a neat trick that we can use here: in the update (6.6) we only need the product $B^{-1}\mathfrak{g}_{\mathbf{u}_k}^f(\mathbf{y}_k, \mathbf{u}_k)$. Thus by the free choice of the metric $B^{-1} = Q_A$, the update (6.6) only depends on the line search oracle $B^{-1}\mathfrak{g}_{\mathbf{u}_k}^f(\mathbf{y}_k, \mathbf{u}_k) = -n\mathtt{LS}^f(\mathbf{y}_k, \mathbf{u}_k)$ and we can indeed implement this scheme. This trick can be applied for all line search oracles $\mathtt{LS}^f$ for which it holds $\mathbb{E}[\mathtt{LS}^f(\mathbf{y}, \mathbf{u})\mathbf{u})] = -Q'_A\nabla f(\mathbf{y})$ for $Q'_A \in \mathrm{PD}_n$ independent of $\mathbf{y}$. However, the difficulty is to estimate the corresponding parameters m and L for the quadratic lower bound (2.11)

and the parameter α that measures the one step progress (6.8) in the B^{-1} norm.

Proof of Corollary 6.5. As announced, we use Theorem 6.11 with $B = Q_A^{-1}$. By Lemma 2.2 we see

$$f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_A^2 \geq \frac{\lambda_{\min}(AQ_A)}{2}\|\mathbf{x}\|_{Q_A^{-1}}^2 ,$$

i.e. for the new metric Q_A^{-1}, the function f is strongly convex for parameter $m = \lambda_{\min}(AQ_A) \geq n^{-1}\kappa_{\mathrm{T}}^{-1}(A)$. The lower bound on $\lambda_{\min}(AQ_A)$ was given in Lemma 4.22 on page 73. As in the proof of Corollary 6.4 above, we now set $\zeta_k = m$ for $k \geq 0$, and use the one step progress to derive a lower bound on α. In Lemma 4.20 we showed that the one step progress for the line search step $\mathtt{LS}^f(\mathbf{y}, \mathbf{u})\mathbf{u}$ is exactly $\frac{1}{2}\|\nabla f(\mathbf{y})\|_{Q_A}^2$. By the assumption (6.2) on the parameter $\omega(A)$ t follows that we can choose α^2 at least as large as $\frac{1}{n\omega(A)}$, that is $\alpha \geq \frac{1}{n^{1/2}\omega^{1/2}(A)}$ and with $\psi_k \leq (1 - \alpha\sqrt{m})^k$, the first inequality follows. For the second inequality, we observe that we derived the upper bound $\omega(A) \leq \kappa_{\mathrm{T}}(A)$ in Lemma 4.23. □

Chapter 7

Conclusion

In this thesis we analyzed the convergence of simple random search schemes. For this, we introduced the framework of Random Pursuit algorithms. These are iterative schemes of the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \sigma_k \mathbf{u}_k \,,$$

where the sequences $(\sigma)_{k\geq 0}$ and $(\mathbf{u}_k)_{k\geq 0}$ denote the step sizes and search directions. The step sizes must be chosen such as to satisfy a sufficient decrease condition. For instance, they can be determined by a line search.

As one of our main contributions, we provided a concise convergence analysis for Random Pursuit algorithms. Our analysis is twofold: (i) we provide *a posteriori* analysis of the convergence of the sequence $(\mathbf{x}_k)_{k\geq 0}$ once the step sizes and search directions have been determined. If the search directions are chosen at random, then we also study the (ii) expected behavior after one, or several iterations. One important instance of a Random Pursuit algorithm is the scheme that selects the search directions independently from the unit sphere, $\mathbf{u}_k \sim S^{n-1}$, and determines the step size by a line search. If the line search oracle satisfies the sufficient decrease condition (D2) from page 48 with parameter γ, then the running time to find an ϵ-approximate solution on a strongly convex function $f \in C^1_{1,1}(A)$ for $A \in \mathrm{PD}_n$ is $O\big(n\kappa_{\mathrm{T}}(A)\frac{1}{\gamma}\ln\frac{1}{\epsilon}\big)$. The quantity $\kappa_{\mathrm{T}}(A)$ can be viewed as a generalization of the condition number $\kappa(A)$, as it depends on the average of the eigenvalues of A instead of only the extremal ones. The dependency of the convergence rate on the full eigenvalue spectrum is most prominently featured in our bounds on

the one step progress, which can be estimated as

$$\mathbb{E}\left[f(\mathbf{x}_{k+1}) - \min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \mid (\mathbf{x}_i)_{i=0}^k\right] \leq (1-\tau_k)\left(f(\mathbf{x}_k) - \min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})\right)$$

for

$$\tau_k = \frac{\gamma}{n\kappa_{\mathrm{E}}(A, I_n, \nabla f(\mathbf{x}_k)} \geq \frac{\gamma}{n\kappa_{\mathrm{T}}(A)} \geq \frac{\gamma}{n\kappa(A)}\,.$$

Here, the best bound depends on κ_{E}. This quantity can be regarded as yet another generalization of the condition number which does not only depend on the spectrum of A, but also on the current position $\mathbf{x}_k$. We conclude that our analysis clearly reveals that the running time (i) depends on the full eigenvalue spectrum of the matrix A and (ii) scales linearly in the dimension n. This dependency can be interpreted in the following way: although the zeroth-order methods are missing any gradient information, they can compensate for this handicap by performing n steps (in random directions), instead of only one step along the gradient direction. Alternatively, the gradient information could also be estimated by finite differences, again at the expense of $O(n)$ additional function evaluations. These observations suggest that our results can easily be generalized to settings where either the optimization is carried out over a $d \leq n$ dimensional subspace (instead of a one-dimensional line search), or to first-order methods that only have access to partial gradient information, say d out of the n entries. In both cases, the running time will supposedly only scale with $\frac{n}{d}$, as opposed to n in case of the pure Random Pursuit considered in this thesis. Furthermore, we observe that the running time depends inversely proportional on the accuracy parameter γ of the line search and conclude that (iii) the quality of the one-dimensional optimization has only a mild effect on the running time. For instance, an accuracy $\gamma = 0.2$ increases the running time only by a factor of 5. This the reason why apparently very simple line search oracles—for instance the adaptive step size scheme (2.7)—are successfully used in many popular randomized search schemes.

In the second part of this thesis we presented a few application of Random Pursuit algorithms, that is, existing algorithmic schemes which can be analyzed in our framework. The most prominent example comprises Kaczmarz' method for solving systems of linear equations. The Random Hessian Estimation scheme of Leventhal and Lewis [141] was of specific interest for two reasons: (i) it admits a simple zeroth-order variable metric scheme—much similar to CMA-ES—that is amenable

to theoretical investigation, as (ii) we revealed that (RHE) is nothing else but Random Pursuit on the space of symmetric matrices for a specific sampling distribution. For this specific instance of Random Pursuit we derived, again, the exact convergence factor and showed tight upper and lower bounds for its convergence. On a quadratic function $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_A^2$, the running time of the scheme that uses first (RHE) to estimate $\hat{A} \approx A$, and then Random Pursuit with search directions $\mathbf{u}_k \sim S_{\hat{A}}^{n-1}$, the running time is $O\big(n^2 \ln \kappa_{\mathrm{F}}(A) + n \ln \frac{1}{\epsilon}\big)$, where $\kappa_{\mathrm{F}}(A)$ is the relative condition number of A. We also present a novel implementation of this scheme which needs only $O\big(n^2 + n \ln \frac{1}{\epsilon}\big)$ function evaluations, independent of A. That is, the scheme is affine invariant. However, when comparing the running time of this approach with the running time of a Random Pursuit without (RHE) and $\mathbf{u}_k \sim S^{n-1}$, we see that the variable metric scheme is only superior if $n \ln \kappa_{\mathrm{F}}(A) \leq \kappa_{\mathrm{T}}(A) \ln \frac{1}{\epsilon}$, or $n \leq \kappa_{\mathrm{T}}(A) \ln \frac{1}{\epsilon}$, respectively. This rules out the (straightforward) application of (RHE) in very high dimensions n. However, if for instance a good approximation $B \approx A$ could be found in linear time, the variable metric approach should be preferred. The approximation B could for instance be of the form $B = R + S$, where R is a low rank matrix and S a sparse matrix, both representable with only $O(n)$ elements. Future research must address the question whether its is possible to modify (RHE) to incorporate such constraints with running time only linear in n, or whether a completely different approximation technique must be used.

The last part of this thesis was dedicated to the study of a completely different acceleration technique. Instead of learning the Hessian matrix and then applying a simple Random Pursuit algorithm, the Accelerated Gradient methods utilize the properties implied by convexity more thoroughly. But the resulting schemes are slightly more complicated. We show that one part of Nesterov's Random Accelerated Gradient method can easily be replaced by a line search. The running time on strongly convex functions $f \in C_{1,1}^1(A)$ for $A \in \mathrm{PD}_n$ reduces to $O\big(n\kappa_{\mathrm{T}}^{1/2}(A)\gamma^{-1/2} \ln \frac{1}{\epsilon}\big)$. Considering the full eigenvalue spectrum of A pays off, as the previously known bounds for the Random Accelerated Gradient method all depend on $\kappa(A)$. The running time still scales linearly in the dimension n, but the dependency on $\kappa_{\mathrm{T}}(A)$ is significantly reduced compared to the simple Random Pursuit. We have to conclude, that the accelerated version of Random Pursuit, denoted as SARP, would be the most suitable scheme from a theoretical point of view, especially in high dimension. However, it is not truly gradient-

free, as it requires access to a gradient oracle. It is the topic of our current research, if and how this dependency can be avoided. As a partial result, we derived the convergence rate for a gradient-free version of SARP on quadratic functions $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_A^2$. The running time $O\big(n\kappa_{\mathrm{T}}^{1/2}(A)\omega^{1/2}(A)\ln\frac{1}{\epsilon}\big)$ depends on a parameter $\omega(A)$, for which no satisfactory bound could be found yet. However, $\omega(A) \leq \kappa_{\mathrm{T}}(A)$, which implies that this version of SARP converges at least as fast as the simple Random Pursuit and hence should be preferred.

We have shown that randomized zeroth-order search schemes can be efficiently applied to convex optimization problems. We have also mentioned two important research questions that would allow to significantly advance our framework: (i) Hessian estimation in linear time and (ii) gradient-free accelerated random search. The first problem is not only limited to Random Pursuit algorithms, but also of specific interest for other branches of zeroth-order optimization. Estimation of the Hessian (or its inverse), is a key ingredient in many schemes, for instance the Evolution Strategies CMA-ES or Gaussian Adaptation. As the (RHE) scheme studied in this thesis is quite different from the schemes used in the latter two algorithms, we do not expect that our presented research can directly be applied to prove the convergence of those schemes. It might be worthwhile to investigate whether it is possible to express the behavior of the Covariance Estimation schemes used in CMA-ES and Gaussian Adaptation in terms of minimization of a (virtual) potential function by an instance of a Random Pursuit algorithm, similar as we did for (RHE). Another, more direct and seemingly more promising way to use our results for the investigation of those two algorithms is the following: Both schemes use essentially a variation of the adaptive step size scheme (2.7) to determine the search steps. If a theoretical analysis of this line search oracle can reveal a non-trivial estimation of accuracy parameter γ (as used in the sufficient decrease condition (D1)), then the search steps of CMA-ES or Gaussian Adaptation for fixed metric, i.e. covariance, can be analyzed in our framework.

In this thesis we restricted our attention to convex functions. Further research will also be concerned with the question whether it is possible to extend some of the results to carefully chosen non-convex problems. For instance, one could investigate the adaptive step size scheme (2.7) on asymptotically convex functions (as defined in Section 2.3.4), or on an appropriately defined subclass of those functions.

Appendix A

Tools and Lemmas

A.1 Selected Random Variables

A.1.1 Normal Random Variables

We here review and derive certain facts about the moments of the standard normal distribution.

Fact A.1 (Moments of Normal Variables)**.** *Let* $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \Sigma)$ *multivariate normal for covariance* $\Sigma \in \mathrm{PD}_n$ *and denote by* $\iota_1, \dots, \iota_{2k}$ *a set of (not necessarily distinct) indices,* $\iota_i \in [n]$*. Then*

$$\mathbb{E}\left[u_{\iota_1} \cdots u_{\iota_{2k-1}}\right] = 0\,, \qquad \mathbb{E}\left[u_{\iota_1} \cdots u_{\iota_{2k}}\right] = \sum \prod \Sigma_{ij}\,,$$

where the notation $\sum \prod$ *means summing over all ways of partitioning* $u_{\iota_1}, \dots, u_{\iota_{2k}}$ *into pairs. Especially, for all indices* i, j, k, l*,*

$$\mathbb{E}[u_i u_j] = \Sigma_{ij}\,, \qquad \mathbb{E}[u_i u_j u_k u_l] = \Sigma_{ij}\Sigma_{kl} + \Sigma_{ik}\Sigma_{jl} + \Sigma_{il}\Sigma_{jk}\,.$$

Proof. The odd moments vanish by symmetry of the normal distribution. The formula for the even moments is known as *Isserlis' Theorem* [110]. □

A.1.2 Products of Quadratic Forms

We consider quadratic forms $Q(\mathbf{u}) = \mathbf{u}^T A \mathbf{u}$ where $\mathbf{u}$ is a normal random vector and $A \in \mathbb{R}^{n \times n}$. Without loss of generality, $A \in \mathrm{SYM}_n$, see Section 2.5.3.

The moments $\mathbb{E}[(\mathbf{u}^T A\mathbf{u})^p]$ of the random variable $(\mathbf{u}^T A\mathbf{u})$ can be expressed in terms of its cumulants (see [156, Thm. 3.3.2]) trough what is called the Faà di Bruno's formula (see the review [45]). Magnus [153] provides exact expressions, Holmquist [105] presents a more direct approach that requires only the enumeration of all permutations of the set $[p]$, and Ghazal gives recursive formulas [69].

Explicit expressions for the product of two (Nagar [170]), three (Neudecker [188]) and four (Kumar [138]) quadratic forms in normal variables were given by several authors (cf. [105]). Kumar [138], Magnus [153] and Holmquist [105] presented algorithmic procedures to generate the expectations of higher order products.

Fact A.2 (Expectation of Products)**.** *Let* $A, B \in \mathrm{SYM}_n$ *and let* $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \Sigma)$*, for covariance* $\Sigma \in \mathrm{PD}_n$*. Then*

$$\begin{aligned}\mathbb{E}\left[\mathbf{u}^T A\mathbf{u}\right] &= \mathrm{Tr}[A\Sigma]\,,\\ \mathbb{E}\left[\mathbf{u}^T A\mathbf{u}\cdot\mathbf{u}^T B\mathbf{u}^T\right] &= \mathrm{Tr}[A\Sigma]\mathrm{Tr}[B\Sigma] + 2\mathrm{Tr}[A\Sigma B\Sigma]\,,\end{aligned}$$

especially, for $B = A$*, we have* $\mathbb{E}\left[(\mathbf{u}^T A\mathbf{u})^2\right] = \mathrm{Tr}[A\Sigma]^2 + 2\mathrm{Tr}[(A\Sigma)^2]$.

Proof. Application of Fact A.1 yields

$$\mathbb{E}\left[\mathbf{u}^T A\mathbf{u}\right] = \sum_{i,j=1}^{n} \mathbb{E}\left[u_i u_j A_{ij}\right] = \sum_{i,j=1}^{n} \Sigma_{ij} A_{ij} = \mathrm{Tr}[A^T\Sigma]\,,$$

and

$$\begin{aligned}\left(\mathbb{E}\left[\mathbf{u}^T A\mathbf{u}\cdot\mathbf{u}\mathbf{u}^T\right]\right)_{ij} &= \sum_{k,l=1}^{n} \mathbb{E}\left[u_i u_j u_k u_l A_{kl}\right]\\ &= \sum_{k,l=1}^{n} A_{kl}\left(\Sigma_{ij}\Sigma_{kl} + \Sigma_{ik}\Sigma_{jl} + \Sigma_{il}\Sigma_{jk}\right)\\ &= \mathrm{Tr}[A\Sigma]\Sigma_{ij} + \left(\Sigma A\Sigma\right)_{ij} + \left(\Sigma A\Sigma\right)_{ji}\,,\end{aligned}$$

using symmetry of $\Sigma A\Sigma$. The claims follows from the observation $\mathbf{u}^T A\mathbf{u}\cdot\mathbf{u}^T B\mathbf{u} = \mathrm{Tr}[\mathbf{u}^T A\mathbf{u}\cdot\mathbf{u}\mathbf{u}^T B]$. □

For $A\Sigma = I_n$, the random variable $\mathbf{u}^T A\mathbf{u}$ follows the χ^2 distribution with n degrees of freedom.

Fact A.3 (χ^2-Moments)**.** *Let* $\mathbf{u} \sim \mathcal{N}(0, A^{-1})$ *with covariance* $A^{-1} \in \mathrm{PD}_n$*. Then*

$$\mathbb{E}[\|\mathbf{u}\|_A^{2k}] = \prod_{i=1}^{k}(n + 2i - 2)\,.$$

Especially, $E[\|\mathbf{u}\|_A^2] = n$ *and* $E[\|\mathbf{u}\|_A^4] = n(n+2)$*.*

Proof. Let $A^{1/2}$ denote the symmetric positive definite root of A. By definition of the normal distribution, the vector $\mathbf{v} = A^{1/2}\mathbf{u}$ for $\mathbf{u} \sim \mathcal{N}(0, A^{-1})$ is normal distributed with covariance $AA^{-1} = I_n$. Thus $\mathbb{E}_{\mathbf{u}}[\|\mathbf{u}\|_A^{2k}] = \mathbb{E}_{\mathbf{v}}[\|\mathbf{v}\|^{2k}]$, which are the moments of a χ^2 distributed random variable. The two special cases follow via this observations from Fact A.2, for the general formula see [62]. □

A.1.3 Ratios of Quadratic Forms

We consider ratios $R_q^p(\mathbf{u}) := Q_A^p/Q_B^q$, in quadratic forms $Q_X(\mathbf{u}) := \mathbf{u}^T X \mathbf{u}$, where $\mathbf{u}$ is a normal random vector and $A \in \mathrm{SYM}_n$ and $B \in \mathrm{PD}_n$ (see Section A.1.2).

Bao and Kan [15] show that the moment $\mathbb{E}[R_q^p]$ exists if and only if $\frac{n}{2} + p > q$. Low order moments have been adressed by De Gooijer [82], building on the work of Sawa [216, 217]. Integral expressions where given in [68, 153, 154, 229] provide exact expressions for all moments. Aspects of numerical evaluation of these expressions are discussed in [194].

We are especially interested in $\mathbb{E}[R_1^1]$ and $\mathbb{E}[R_2^2]$. For exact expressions see [68, 121]. If Q_A/Q_B is independent of Q_B, then $\mathbb{E}[R_p^p] = \mathbb{E}[Q_A^p]/\mathbb{E}[Q_B^p]$, [100, 196]. This holds for elliptical random variables.

Lemma A.4 (Quadratic forms of elliptical variables)**.** *Let* $\Sigma \in \mathrm{PD}_n$*,* $A, B \in \mathrm{SYM}_n$ *and let* $\mathbf{u} \sim S_{\Sigma^{-1}}^{n-1}$ *elliptically distributed for metric* Σ^{-1}*. Then*

$$\begin{aligned}\mathbb{E}\left[\mathbf{u}^T A \mathbf{u}\right] &= \tfrac{\mathrm{Tr}[A\Sigma]}{n}\,,\\ \mathbb{E}\left[\mathbf{u}^T A \mathbf{u} \cdot \mathbf{u}^T B \mathbf{u}^T\right] &= \tfrac{\mathrm{Tr}[A\Sigma]\mathrm{Tr}[B\Sigma] + 2\mathrm{Tr}[A\Sigma B\Sigma]}{n(n+2)}\,,\end{aligned}$$

especially, for $B = A$*, we have* $\mathbb{E}\left[(\mathbf{u}^T A \mathbf{u})^2\right] = \frac{\mathrm{Tr}[A\Sigma]^2 + 2\mathrm{Tr}[(A\Sigma)^2]}{n(n+2)}$*.*

Proof. Let $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \Sigma)$. The random vector $\mathbf{w} = \mathbf{v}/\|\mathbf{v}\|_{\Sigma^{-1}}$ has the same distribution as $\mathbf{u}$ by definition, see Remark 2.5. Note that the

ratio $R(\frac{\mathbf{v}}{\|\mathbf{v}\|_{\Sigma^{-1}}}) := \frac{\mathbf{v}^T A\mathbf{v}\cdot\mathbf{v}^T B\mathbf{v}^T}{\|\mathbf{v}\|^4_{\Sigma^{-1}}}$ is independent of $\|\mathbf{v}\|^4_{\Sigma^{-1}}$ (R does only depend on the direction of $\mathbf{v}$). In particular,

$$\mathbb{E}_{\mathbf{u}}\left[\mathbf{u}^T A\mathbf{u}\cdot\mathbf{u}^T B\mathbf{u}^T\right] = \mathbb{E}_{\mathbf{v}}\left[\frac{\mathbf{v}^T A\mathbf{v}\cdot\mathbf{v}^T B\mathbf{v}^T}{\|\mathbf{v}\|^4_{\Sigma^{-1}}}\right] = \frac{\mathbb{E}_{\mathbf{v}}\left[\mathbf{v}^T A\mathbf{v}\cdot\mathbf{v}^T B\mathbf{v}^T\right]}{\mathbb{E}_{\mathbf{v}}\left[\|\mathbf{v}\|^4_{\Sigma^{-1}}\right]},$$

see [100, 196]. The values of the numerator and denumerator were given in Fact A.2 and Fact A.3. The analogous reasoning applies for the first moment. □

Remark A.5 (Covariance of elliptical variables)**.** *By the same reasoning, one obtains* $\mathbb{E}[\mathbf{u}\mathbf{u}^T] = \frac{1}{n}\Sigma$ *for* $\mathbf{u} \sim S^{n-1}_{\Sigma^{-1}}$ *and* $\Sigma \in \mathrm{PD}_n$*.*

A.1.4 Scaled Normal and Elliptical Vectors

Lemma A.6 (Scaled normal vectors)**.** *Let* $\mathbf{u} \in \mathcal{N}(\mathbf{0}, \Sigma)$ *with* $\Sigma \in \mathrm{PD}_n$*, let* $C, D \in \mathrm{SYM}_n$ *and* $\mathbf{x} \in \mathbb{R}^n$*. Then*

$$\begin{aligned}
\mathbb{E}\left[\langle\mathbf{x},\mathbf{u}\rangle\,\mathbf{u}\right] &= \Sigma\mathbf{x}\,,\\
\mathbb{E}\left[\langle D\mathbf{x},\mathbf{u}\rangle\,\langle\mathbf{x},\mathbf{u}\rangle\right] &= \mathbf{x}^T(D\Sigma)\mathbf{x} = \|\mathbf{x}\|^2_{D\Sigma}\,,\\
\mathbb{E}\left[\|\langle D\mathbf{x},\mathbf{u}\rangle\,\mathbf{u}\|^2_C\right] &= \mathrm{Tr}[C\Sigma]\,\|\mathbf{x}\|^2_{D\Sigma D} + 2\,\|\mathbf{x}\|^2_{D\Sigma C\Sigma D}\,.
\end{aligned}$$

Proof. By Fact A.1 and linearity of expectation we deduce

$$\mathbb{E}\left[\langle\mathbf{x},\mathbf{u}\rangle\,\mathbf{u}\right] = \mathbb{E}\left[\mathbf{u}\mathbf{u}^T\mathbf{x}\right] = \mathbb{E}\left[\mathbf{u}\mathbf{u}^T\right]\mathbf{x} = \Sigma\mathbf{x}\,,$$

The second claim follows from $\langle B\mathbf{x},\mathbf{u}\rangle\,\langle\mathbf{x},\mathbf{u}\rangle = \mathbf{x}^T B\,(\langle\mathbf{x},\mathbf{u}\rangle\,\mathbf{u})$. The last moment follows from Fact A.2 with $A = C$ and $B = (D\mathbf{x})(D\mathbf{x})^T$. Observe $\mathrm{Tr}[(D\mathbf{x})(D\mathbf{x})^T\Sigma] = \mathbf{x}^T(D^T\Sigma D)\mathbf{x}$ and $\mathrm{Tr}[C\Sigma(D\mathbf{x})(D\mathbf{x})^T\Sigma] = \mathbf{x}^T(D^T\Sigma C\Sigma D)\mathbf{x}$. □

Analogous to Lemma A.6, we can immediately compute the same expectations for elliptical random variables.

Lemma A.7. *Let* $\mathbf{u} \sim S^{n-1}_{\Sigma^{-1}}$ *for metric* $\Sigma \in \mathrm{PD}_n$ *and* $C, D \in \mathrm{SYM}_n$ *and* $\mathbf{x} \in \mathbb{R}^n$*. Then*

$$\begin{aligned}
\mathbb{E}\left[\langle\mathbf{x},\mathbf{u}\rangle\,\mathbf{u}\right] &= \tfrac{1}{n}\Sigma\mathbf{x}\,,\\
\mathbb{E}\left[\langle D\mathbf{x},\mathbf{u}\rangle\,\langle\mathbf{x},\mathbf{u}\rangle\right] &= \tfrac{1}{n}\,\|\mathbf{x}\|^2_{D\Sigma}\,,\\
\mathbb{E}\left[(\mathbf{u}^T D\mathbf{u})^2\right] &= \tfrac{\mathrm{Tr}[D\Sigma]^2 + 2\mathrm{Tr}[(D\Sigma)^2]}{n(n+2)}\,,\\
\mathbb{E}\left[\|\langle D\mathbf{x},\mathbf{u}\rangle\,\mathbf{u}\|^2_C\right] &= \tfrac{\mathrm{Tr}[C\Sigma]\|\mathbf{x}\|^2_{D\Sigma D} + 2\|\mathbf{x}\|^2_{D\Sigma C\Sigma D}}{n(n+2)}\,.
\end{aligned}$$

A.2 Ratio of Quadratic Forms

Proof of Lemma 2.2. Let $B^{1/2} \in \mathrm{PD}_n$ be the positive definite root of B, and set $\mathbf{y} = B^{1/2}\mathbf{x}$. Then we have

$$\frac{\|\mathbf{x}\|_A^2}{\|\mathbf{x}\|_B^2} = \frac{\|\mathbf{y}\|_{B^{-1/2}AB^{-1/2}}^2}{\|\mathbf{y}\|^2}\,.$$

Max- and minimizing the above ratio for $\|\mathbf{y}\| = 1$, and the bounds from (2.8), yield

$$\lambda_{\min}(B^{-1/2}AB^{-1/2}) \leq \|\mathbf{y}\|_{B^{-1/2}AB^{-1/2}}^2 \leq \lambda_{\max}(B^{-1/2}AB^{-1/2})\,.$$

The inequalities follow by the fact that the matrices $B^{-1/2}AB^{-1/2}$ and $AB^{-1/2}B^{-1/2} = AB^{-1}$ have the same eigenvalues, see e.g. [204, Prop. 13.2]. The inequalities in (2.8) are tight, thus we have equality for $\mathbf{x} = \mathbf{v}_{\min}$ and $\mathbf{x} = \mathbf{v}_{\max}$, the eigenvectors corresponding the the minimal and maximal eigenvalue of AB^{-1}, respectively. □

A.3 Perturbation

Lemma A.8 (Perturbation)**.** *Let $A \in \mathrm{PD}_n$, $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{z}_1 \in \mathbb{R}^n$ an eigenvector corresponding to the smallest eigenvalue of $(A - \mathbf{x}\mathbf{x}^T)$. Then*

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

Proof. The matrix $(A - \mathbf{x}\mathbf{x}^T)$ is symmetric. Hence, its spectral decomposition $(A - \mathbf{x}\mathbf{x}^T) = \sum_{i=1}^n \lambda_i \mathbf{z}_i\mathbf{z}_i^T$ with $\lambda_1 \leq \lambda_2 \leq \ldots \lambda_n$ in increasing order exists. If $\lambda_1 \geq 0$, then there is nothing to show. Otherwise, we observe that by a variant of Weyl's theorem (cf. [107, 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 λ_1 can be negative. We conclude

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

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

A.4 Slow Convergence with Additive Error

Lemma A.9. *Let $(f_k)_{k\geq 1}$ be a sequence of positive numbers. Suppose*

$$f_{k+1} \leq (1-\theta/k)\, f_k + C\theta^2/k^2 + D\,, \quad \text{for } k \geq 1,$$

for constants $\theta > 1$, $C > 0$ and $D \geq 0$. Then it follows by induction that

$$f_k \leq Q(\theta)/k + (k-1)D\,,$$

where $Q(\theta) = \max\left\{\theta^2 C/(\theta-1), f_1\right\}$.

A very similar result was stated without proof in [177] and also Hazan [99] is using the same.

Proof. For $k = 1$ it holds that $f_k \leq Q(\theta)$ by definition of $Q(\theta)$. Assume that the result holds for $k \geq 1$. If $Q(\theta) = \theta^2 C/(\theta-1)$ then we deduce:

$$\begin{aligned} f_{k+1} &\leq \frac{\theta^2 C(k-\theta)}{(\theta-1)k^2} + \frac{C\theta^2}{k^2} + \frac{(k-\theta)(k-1)D}{k} + D \\ &= \frac{\theta^2 C(k-1)}{(\theta-1)k^2} + \frac{D(k^2-\theta(k-1))}{k} \leq \frac{\theta^2 C}{(\theta-1)(k+1)} + kD\,. \end{aligned}$$

If on the other hand $Q(\theta) = f_1$, then

$$f_1 \geq \frac{\theta^2 C}{(\theta-1)} \quad \Leftrightarrow \quad (\theta-1)f_1 \geq \theta^2 C\,,$$

and it follows

$$\begin{aligned} f_{k+1} &\leq \frac{(k-\theta)f_1}{k^2} + \frac{C\theta^2}{k^2} + \frac{(k-\theta)(k-1)D}{k} + D \\ &= \frac{(k-1)f_1}{k^2} + \frac{\theta^2 C - (\theta-1)f_1}{k^2} + \frac{D(k^2-\theta(k-1))}{k} \\ &\leq \frac{f_1}{k+1} + kD\,. \end{aligned}$$

□

Appendix B

Deferred Proofs

B.1 Convergence with Sufficient Decrease

Proof of Theorem 3.3 (ii)-(iii). Let us start with the part (ii), where $\epsilon_k = 0$. This case is similar to the one treated in [264]. By convexity (2.9) and the assumptions on the diameter R we have

$$f_k \leq \langle \nabla f(\mathbf{x}_k), \mathbf{x}^* - \mathbf{x}_k \rangle \leq R \, \|\nabla f(\mathbf{x}_k)\|_{A^{-1}} \ ,$$

for $k = 0, \ldots, N-1$. By combining this lower bound on $\|\nabla f(\mathbf{x}_k)\|_{A^{-1}}$ with the sufficient decrease condition (D1) we deduce

$$f_k - f_{k+1} \geq \frac{\gamma_k \beta_k^2}{2LR^2} f_k^2 \, ,$$

for $k = 0, \ldots, N-1$. Let $\tau_k := \gamma_k \beta_k^2 / (2LR^2)$. We claim that the sequence $(f_k)_{k \geq 0}$ satisfies $f_{k+1}^{-1} - f_k^{-1} \geq \tau_k$. Indeed,

$$\frac{1}{f_{k+1}} - \frac{1}{f_k} = \frac{f_k - f_{k+1}}{f_k f_{k+1}} \geq \frac{\tau_k f_k}{f_{k+1}} \geq \tau_k \, ,$$

where the last inequality follows from the simple observation, that the fraction $f_k / f_{k+1} \geq 1$ if $\tau_k \geq 0$. By summing the established inequality over all $k = 0, \ldots, N-1$ we deduce

$$\frac{1}{f_N} - \frac{1}{f_0} = \sum_{k=0}^{N-1} \left(\frac{1}{f_{k+1}} - \frac{1}{f_k} \right) \geq S_N \, .$$

The statement follows by rearranging the terms in the derived inequality and the observation $f_0 > 0$ if $S_N > 0$.

Now we proceed to part (iii). We begin similarly as above. The assumptions and the sufficient decrease condition (D1) yield

$$f_k - f_{k+1} + \epsilon \geq \tau f_k^2 \,, \tag{B.1}$$

for $k = 0, \ldots, N-1$ and $\tau := \delta/(2LR^2)$. We rewrite this bound on f_{k+1} as

$$f_{k+1} - \epsilon \leq f_k - \tau f_k^2 = f_k + 2 \min_{h_k} \left(-h_k f_k + \frac{h_k^2}{2\tau} \right) \leq (1 - 2h_k) f_k + \frac{h_k^2}{\tau} \,,$$

where the last inequality holds for arbitrary parameter h_k. Therefore

$$f_{k+1} \leq (1 - 2h_k) + h_k^2 \frac{2LR^2}{\delta} + \epsilon \,.$$

By setting $h_k = 1/(k+1)$ for $k = 0, \ldots, N-1$, we obtain a recurrence that is exactly of the form as treated in Lemma A.9 on page 116. □

B.2 Interpolation of Quadratic Functions

Proof of Example 4.5. The function $g(t) := f(\mathbf{x} + (s+t)\mathbf{u})$ is quadratic and can be written as

$$g(t) = f(\mathbf{x} + s\mathbf{u}) + t\left(sf(\mathbf{u}) + \mathbf{x}^T A \mathbf{u}\right) + t^2 f(\mathbf{u}) \,. \tag{B.2}$$

Observe

$$a = \frac{g(\epsilon) - 2g(0) + g(-\epsilon)}{\epsilon^2} = \frac{2\epsilon^2 f(\mathbf{u})}{\epsilon^2} = 2f(\mathbf{u}) \,,$$

and

$$b = \frac{g(\epsilon) - g(-\epsilon)}{2\epsilon} = \frac{2\epsilon\left(sf(\mathbf{u}) + \mathbf{x}^T A \mathbf{u}\right)}{2\epsilon} = sf(\mathbf{u}) + \mathbf{x}^T A \mathbf{u} \,.$$

Therefore (B.2) can equivalently be written as $g(t) = f_s + bt + \frac{a}{2}t^2$. By setting the derivative of g to zero, $\nabla g(t) = b + at \stackrel{!}{=} 0$, we see that g attains its minima for $t = -\frac{b}{a}$. □

B.3 Typical Search Position

Theorem 3.6 on page 45 depends on the expectation of the squared angle measure $\mathbb{E}[\beta_k^2]$ for every step k. The important lower bound derived in Example 4.12 on page 65 depends on $\kappa_{\mathrm{E}}(A, B, \mathbf{x}_k)$, where A, B are two fixed matrices, and $\mathbf{x}_k \in \mathbb{R}^n$ the current search point. This makes it hard to derive a uniform lower bound on $\mathbb{E}[\beta_k^2]$, that does not depend on k. The following remark shows that instead of considering every single $\mathbf{x}_k$ for $k = 1, \ldots, N$, it is enough to consider an "average" point $\bar{\mathbf{x}} \in \mathbb{R}^n$ instead. This point can be viewed as a "typical" position of the scheme with respect to the level sets of the objective function, see the discussion in Section 4.3.1.

Remark B.1. *Consider the same setting as in Theorem 3.6 and assume* $\mathbb{E}[\gamma_k \beta_k^2] \geq \frac{a}{\sigma(\mathbf{x}_k)+b}$, *for* $a, b > 0$ *constant and* $\sigma \colon \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ *nonnegative. Then*

$$\mathbb{E}[f_N] \leq f_0 \cdot \exp\left[-\frac{amN}{L(\bar{\sigma}+b)}\right] + D_N\,, \tag{B.3}$$

where $\bar{\sigma} = \frac{1}{N}\sum_{i=0}^{N-1} \sigma(\mathbf{x}_k)$.

Proof. By Theorem 3.6 it suffices to consider the expectation of the sum $S_N := \sum_{k=0}^{N-1} \gamma_k \beta_k^2$. The function $\frac{1}{x}$ is convex, therefore by Jensen's inequality and the assumptions we find

$$\mathbb{E}[S_N] \geq \sum_{k=0}^{N-1} \frac{a}{\sigma(\mathbf{x}_k)+b} \geq \frac{aN}{\frac{1}{N}\sum_{i=0}^{N-1}\sigma(\mathbf{x}_k)+b} = \frac{aN}{\bar{\sigma}+b}\,,$$

and the claim follows from monotonicity of e^x. □

B.4 Weighted Sampling of a Discrete Set

Proof of Example 4.15. If C has full rank n, then the matrix $C^TC \in \mathbb{R}^{n\times n}$ has also full rank. This well-known property of Gram matrices can be shown in the following way: let $\mathbf{x} \in \mathbb{R}^n$ s.t. $C^TC\mathbf{x} = 0$. Then $0 = \mathbf{x}^TC^TC\mathbf{x} = \|C\mathbf{x}\|^2$. Thus the null space of C^TC is contained in the null space of C, which is trivial by assumption. Thus the inverse $(C^TC)^{-1}$ exists and we can set $C^{-1} := (C^TC)^{-1}C^T$ the left inverse of C. For the second part, note that

$$E_{\mathbf{u}\sim_{wT}}[\mathbf{u}\mathbf{u}^T] = \sum_{i=1}^{m} \frac{\|\mathbf{c}_i\|^2}{\sum_{j=1}^{m}\|\mathbf{c}_j\|^2} \frac{\mathbf{c}_i\mathbf{c}_i^T}{\|\mathbf{c}_i\|^2} = \frac{1}{\|C\|_F^2}\sum_{i=1}^{m}\mathbf{c}_i\mathbf{c}_i^T = \frac{C^TC}{\|C\|_F^2}\,.$$

In Fact B.2 (just below) we derive $\lambda_{\min}(C^TC) \geq \left\|C^{-1}\right\|_2^{-2}$ and the statement follows from the Lemma 4.8. □

Fact B.2 (see e.g. [204])**.** *Let $C \in \mathbb{R}^{m\times n}$ and $\mathbf{x} \in \mathbb{R}^n$. If the left inverse $C^{-1} := (C^TC)^{-1}C^T$ with $C^{-1}C = I_n$ exists, then*

$$\left\|C^{-1}\right\|^{-2} = \lambda_{\min}(C^TC)\,, \quad \textit{especially,} \quad \|\mathbf{x}\|^2 \leq \left\|C^{-1}\right\|^2 \|\mathbf{x}\|^2_{C^TC}\,.$$

Proof. We note $\left\|C^{-1}\right\|^2 = \left\|(C^TC)^{-1}C^TC(C^TC)^{-1}\right\| = \left\|(C^TC)^{-1}\right\|$, which by definition equals the maximal eigenvalue of $(C^TC)^{-1}$, i.e. $\lambda^{-1}_{\min}(C^TC)$. The inequality follows with Lemma 2.2 by setting $B = C^TC$ (and $A = I_n$). □

B.5 Approximating the Covariance Matrix

Proof of Example 4.18. Let $\hat{\Sigma} = \mathbb{E}_{\mathbf{u}\sim T}$ the sample covariance and $\Sigma = \mathbb{E}[\mathbf{u}\mathbf{u}^T] = \frac{1}{n}I_n$ the exact covariance (see Example 4.9). By Fact 4.17 the sample covariance approximates Σ up to a factor of ϵ, that is, equation (4.8) implies $\hat{\Sigma} \succeq (1-\epsilon)\Sigma$, and the statement follows from Lemma 4.8. □

Proof of Example 4.19. For $Y \in \mathrm{SYM}_n$ and $U = \mathbf{u}\mathbf{u}^T$ for $\mathbf{u} \in S^{n-1}$ we have $\langle Y, U\rangle^2 = \sum_{ijkl} u_iu_ju_ku_l\alpha_{ijkl}(Y)$ for coefficients $\alpha_{ijkl}(Y)$. Adamczak et al. [3, Thm. 4.2] prove that a quadratic $\Theta(n^2)$ number of i.i.d. samples from S^{n-1} are enough to approximate its fourth marginal moments (as in equation (4.7) on page 68) with sufficient accuracy. Hence, all the $u_iu_ju_ku_l$ terms in the above sum can also be approximated by the sample estimation. These calculations are detailed in [237, Thm. 6]. □

B.6 Exact One Step Progress

Proof of Lemma 4.20. Example 4.5 gives an analytic expression for $\mathtt{LS}^f$ on a quadratic function. For $\mathbf{x}, \mathbf{u} \in \mathbb{R}^n$, we have:

$$\mathtt{LS}^f(\mathbf{x},\mathbf{u})\mathbf{u} = -\frac{\mathbf{x}^TA\mathbf{u}}{2f(\mathbf{u})}\mathbf{u} = -\frac{\mathbf{u}\mathbf{u}^T}{\|\mathbf{u}\|_A^2}\nabla f(\mathbf{x})\,. \tag{B.4}$$

Therefore, we see that $Q_A = \mathbb{E}\left[\frac{\mathbf{u}\mathbf{u}^T}{\|\mathbf{u}\|_A^2}\right]$ was exactly defined to describe the the expected step $\mathbb{E}[\mathtt{LS}^f(\mathbf{x},\mathbf{u})\mathbf{u}] = -Q_A\nabla f(\mathbf{x})$. For fixed $\mathbf{u}$ and

quadratic f we have:

$$f(\mathbf{x}_+) = f(\mathbf{x}) + \left\langle \nabla f(\mathbf{x}), \mathtt{LS}^f(\mathbf{x},\mathbf{u})\mathbf{u} \right\rangle + \frac{1}{2}\left\|\mathtt{LS}^f(\mathbf{x},\mathbf{u})\mathbf{u}\right\|_A^2$$
$$= f(\mathbf{x}) - \frac{1}{2}\left\langle \nabla f(\mathbf{x}), \frac{\mathbf{u}\mathbf{u}^T}{\|\mathbf{u}\|_A^2}\nabla f(\mathbf{x}) \right\rangle ,$$

where we used the expression (B.4) for $\mathtt{LS}^f$. The claim follows by taking expectation on both sides. □

Proof of Lemma 4.22. The lemma follows by comparing two different bounds on the one step progress $f(\mathbf{x}) - \mathbb{E}[f(\mathbf{x}_+) \mid \mathbf{x}]$. In Lemma 4.20 we quantified the exact one step progress as $\frac{1}{2}\|\nabla f(\mathbf{x})\|_{Q_A}^2$, and in Example 4.12 we derived the lower bound $\frac{1}{2n\kappa_{\mathrm{T}}(A)}\|\nabla f(\mathbf{x})\|_{A^{-1}}^2$. Therefore

$$\frac{\|\nabla f(\mathbf{x})\|_{A^{-1}}^2}{\|\nabla f(\mathbf{x})\|_{Q_A}^2} \leq n\kappa_{\mathrm{T}}(A)\,.$$

On the other hand, by Lemma 2.2 we know that $\lambda_{\min}^{-1}(AQ_A)$ is the smallest number a such that $\|\nabla f(\mathbf{x})\|_{A^{-1}}^2 \leq a\|\nabla f(\mathbf{x})\|_{Q_A}^2$ holds. □

B.7 Matrix Valued Random Pursuit

Proof of Remark 5.2. Without loss of generality assume $\|U_k\|_F = 1$. To show (i), we calculate the scalar product

$$\langle X_{k+1}, U_k\rangle = \langle X_k, U_k\rangle - \langle X_k, U_k\rangle \langle U_k, U_k\rangle = 0\,.$$

Next, (ii) follows by calculating the Frobenius norm of X_{k+1} explicitly:

$$\|X_{k+1}\|_F^2 = \langle X_{k+1}, X_{k+1}\rangle = \langle X_k, X_k\rangle - 2\langle X_k, U_k\rangle\langle U_k, X_k\rangle + \langle X_k, U_k\rangle^2$$
$$= \|X_k\|_F^2 - \langle X_k, U_k\rangle^2$$

Finally, to show (iii), we verify:

$$PX_{k+1}P^T = P\left(X_k - \langle X_k, U_k\rangle U_k\right)P^T$$
$$= PX_kP^T - \langle I_nX_k, U_kI_n\rangle PU_kP^T$$
$$= PX_kP^T - \left\langle PX_kP^T, PU_kP^T\right\rangle PU_kP^T\,,$$

where the last line follows from the cyclic-shift property of the trace: $\mathrm{Tr}[X_kI_nU_kP^TP] = \mathrm{Tr}[PX_kI_nU_kP^T] = \mathrm{Tr}[PX_kP^TPU_kP^T]$. □

Lemma B.3 (Matrix diagonalization)**.** *Let $n \geq 1$ and consider the following 2×2 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. □

Lemma B.4. *Let λ_1, λ_2 as in Theorem 5.4 and $n \geq 2$. Then*

$$\begin{aligned} 1-\frac{2}{n} \leq &\lambda_1 \leq 1-\frac{2}{n+1}, \\ 1-\frac{5}{2n(n+2)} \leq &\lambda_2 \leq 1-\frac{2}{n(n+2)}. \end{aligned}$$

Proof. The main inequality we use $2n \leq \omega \leq (2n+1)$ for $n \geq 2$. Thus

$$\begin{aligned} \lambda_1 &\leq \frac{2n^2-5}{2n(n+2)} = \frac{(n+1)(2n^2-5)}{2n(n+1)(n+2)} \leq \frac{2(n-1)n(n+2)}{2n(n+1)(n+2)} = 1-\frac{2}{n+1}, \\ \lambda_1 &\geq \frac{n^2-3}{n(n+2)} \geq \frac{(n-2)(n+2)}{n(n+2)} = 1-\frac{2}{n}. \end{aligned}$$

Similarly for the last two inequalities:

$$\begin{aligned} \lambda_2 &\leq \frac{n^2+2n-2}{n(n+2)} = 1-\frac{2}{n(n+2)}, \\ \lambda_2 &\geq \frac{2n^2+4n-5}{2n(n+2)} \geq \frac{n^2+2n-5/2}{n(n+2)} = 1-\frac{5}{2n(n+2)}. \end{aligned}$$

□

B.8 Bound on the Convergence Factor

Proof of Lemma 5.6. To show the lemma, we derive bounds on the spectral norm of AB_{K+j}^{-1}. Let $E_k := B_k - A$. The sequence $(E_k)_{k\geq 0}$ is a Local Search Scheme of the form (5.3) and by part (iii) of Corollary 5.3 we can estimate

$$\mathbb{E}[\|E_{K+j}\|_F^2] \leq \eta^j \cdot \|E_0\|_F^2 \cdot e^{-\eta K} \leq \eta^j \cdot \frac{\eta}{b\,\|A^{-1}\|^2} < \frac{\eta^j}{b\,\|A^{-1}\|_2^2}\,.$$

Thus by the reasoning of Lemma 3.15, we have $\|E_{K+j}\|_F^2 < \eta^j \cdot \|A^{-1}\|_2^{-2}$ with probability at least $\left(1 - \frac{1}{b}\right)$. If this inequality holds, then also $\|A^{-1}\|_2 \|E_{K+j}\|_2 \leq \|A^{-1}\|_2 \|E_{K+j}\|_F < \eta^j$. The two claims of the lemma now follow by Lemma B.5 below. □

Lemma B.5 (Convergence factor)**.** *Let $A \in \mathrm{PD}_n$ and let $E \in \mathrm{SYM}_n$ with $\|A^{-1}\|\,\|E\|_2 \leq c$ for $c < 1$ and ϱ_A as in* (5.1)*. Then*

$$\varrho_A(E + A) \leq 1 - \frac{1 - c}{(1 + c)n}\,.$$

Proof. To show the lemma, we derive bounds on the smallest and largest eigenvalues of the matrix $(E + A)A^{-1}$. First we observe that

$$\max\left(\left|\lambda_{\min}(EA^{-1})\right|, \left|\lambda_{\max}(EA^{-1})\right|\right) = \|EA^{-1}\|_2 \leq \|E\|_2 \|A^{-1}\|_2 < c\,,$$

by definition of the spectral norm and submultiplicativitiy. Therefore, $\lambda_{\min}((E + A)A^{-1}) = \lambda_{\min}(I_n + EA^{-1}) \geq 1 - c > 0$, and $(E + A)A^{-1} \in \mathrm{PD}_n$. We have

$$\kappa\big((E + A)A^{-1}\big) = \frac{\lambda_{\min}(I_n + EA^{-1})}{\lambda_{\max}(I_n + EA^{-1})} \geq \frac{1 - c}{1 + c}\,. \qquad \square$$

B.9 Estimate Sequence Construction

Proof of Lemma 6.10. Adapted from Lee and Sidford [140]. We prove the form of ϕ_k by induction. Suppose that $\phi_k(\mathbf{x}) = \Phi_k^* + \frac{\zeta_k}{2}\|\mathbf{x} - \mathbf{v}_k\|_B^2$. By the update rule, we see that

$$\nabla^2\phi_{k+1}(\mathbf{x}) = (1 - \alpha_k)\nabla^2\phi_k(\mathbf{x}) + \alpha_k mB = [(1 - \alpha_k)\zeta_k + \alpha_k m]\,B\,.$$

Therefore, $\phi_{k+1}(\mathbf{x}) = \Phi^*_{k+1} + \frac{\zeta_{k+1}}{2}\|\mathbf{x} - \mathbf{v}_{k+1}\|_B^2$ for ζ_{k+1} and Φ^*_{k+1} and $\mathbf{v}_{k+1}$ yet to be determined. To compute $\mathbf{v}_{k+1}$, we compute the derivative of ϕ_{k+1} and note

$$\begin{aligned}\nabla\phi_{k+1}(\mathbf{x}) &= (1-\alpha_k)\zeta_k B(\mathbf{x}-\mathbf{v}_k) + \alpha_k \mathfrak{g}^f_{\mathbf{u}_k}(\mathbf{y}_k) + \alpha_k m B(\mathbf{x}-\mathbf{y}_k)\\ &= \zeta_{k+1}B\mathbf{x} - (1-\alpha_k)\zeta_k B\mathbf{v}_k + \alpha_k \mathfrak{g}^f_{\mathbf{u}_k}(\mathbf{y}_k) - \alpha_k m B\mathbf{y}_k\,.\end{aligned}$$

On the other hand,

$$\nabla\phi_{k+1}(\mathbf{x}) = \zeta_{k+1}B(\mathbf{x}-\mathbf{v}_{k+1})\,.$$

Combining these two expressions for $\nabla\phi_{k+1}(\mathbf{x})$ and applying B^{-1} on both sides yields the desired formula for $\mathbf{v}_{k+1}$.

Finally, to compute Φ^*_{k+1}, we evaluate $\phi_{k+1}(\mathbf{y}_k)$ in two different ways. First, by the update rule we have

$$\begin{aligned}\Phi^*_{k+1} + \frac{\zeta_{k+1}}{2}\|\mathbf{y}_k - \mathbf{v}_{k+1}\|_B^2 &= (1-\alpha_k)\phi_k(\mathbf{y}_k) + \alpha_k f(\mathbf{y}_k)\\ &= (1-\alpha_k)\left[\Phi^*_k + \frac{\zeta_k}{2}\|\mathbf{y}_k - \mathbf{v}_k\|_B^2\right]\\ &\quad + \alpha_k f(\mathbf{y}_k)\,. \qquad \text{(B.5)}\end{aligned}$$

On the other hand, by the form of $\mathbf{v}_{k+1}$, we have

$$\begin{aligned}\zeta_{k+1}^2\|\mathbf{v}_{k+1} - \mathbf{y}_k\|_B^2 &= \|(1-\alpha_k)\zeta_k\mathbf{v}_k + (\alpha_k m - \zeta_{k+1})\mathbf{y}_k\\ &\quad -\alpha_k B^{-1}\mathfrak{g}^f_{\mathbf{u}_k}(\mathbf{y}_k)\|_B^2\\ &= \left\|(1-\alpha_k)\zeta_k(\mathbf{v}_k - \mathbf{y}_k) - \alpha_k B^{-1}\mathfrak{g}^f_{\mathbf{u}_k}(\mathbf{y}_k)\right\|_B^2\\ &= (1-\alpha_k)^2\zeta_k^2\|\mathbf{v}_k - \mathbf{y}_k\|_B^2 + \frac{\alpha_k^2}{2\zeta_{k+1}}\left\|\mathfrak{g}^f_{\mathbf{u}_k}(\mathbf{y}_k)\right\|_{B^{-1}}^2\\ &\quad - 2(1-\alpha_k)\alpha_k\zeta_k\left\langle \mathfrak{g}^f_{\mathbf{u}_k}(\mathbf{y}_k), \mathbf{v}_k - \mathbf{y}_k\right\rangle\,. \qquad \text{(B.6)}\end{aligned}$$

Combining (B.5) and (B.6), together with

$$\begin{aligned}(1-\alpha_k)\frac{\zeta_k}{2} - \frac{\zeta_{k+1}}{2}\cdot\frac{(1-\alpha_k)^2\zeta_k^2}{\zeta_{k+1}^2} &= \frac{(1-\alpha_k)\zeta_k}{2\zeta_{k+1}}\left[\zeta_{k+1} - (1-\alpha_k)\zeta_k\right]\\ &= \frac{m\alpha_k(1-\alpha_k)\zeta_k}{2\zeta_{k+1}}\,,\end{aligned}$$

yields the desired form of Φ^*_{k+1}. □

Bibliography

[1] M. Abramson and C. Audet. Convergence of Mesh Adaptive Direct Search to second-order stationary points. *SIAM Journal on Optimization*, 17(2):606–619, 2006.

[2] M. Abramson, C. Audet, J. Dennis, and S. Digabel. OrthoMADS: A deterministic MADS instance with orthogonal directions. *SIAM Journal on Optimization*, 20(2):948–966, 2009.

[3] R. Adamczak, A. E. Litvak, A. Pajor, and N. Tomczak-Jaegermann. Quantitative estimates of the convergence of the empirical covariance matrix in log-concave ensembles. *Journal of the AMS*, 23:535–561, 2010.

[4] Y. Akimoto. Analysis of a Natural Gradient algorithm on monotonic convex-quadratic-composite functions. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, GECCO '12, pages 1293–1300, New York, NY, USA, 2012. ACM.

[5] Y. Akimoto, Y. Nagata, I. Ono, and S. Kobayashi. Bidirectional relation between CMA evolution strategies and natural evolution strategies. In R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, editors, *Parallel Problem Solving from Nature, PPSN XI*, volume 6238 of *LNCS*, pages 154–163. Springer Berlin Heidelberg, 2010.

[6] C. Andrieu and J. Thoms. A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4):343–373, 2008.

[7] L. Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3, 1966.

[8] D. V. Arnold and N. Hansen. A (1+1)-CMA-ES for constrained optimisation. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, GECCO '12, pages 297–304, New York, NY, USA, 2012. ACM.

[9] R. B. Ash. *Probability and Measure Theory*. Academic Press, Inc., 2000.

[10] G. Aubrun. Sampling convex bodies: a random matrix approach. *Proceedings of the American Mathematical Society*, 135:1293–1303, 2007.

[11] C. Audet and D. Orban. Finding optimal algorithmic parameters using derivative-free optimization. *SIAM Journal on Optimization*, 17(3):642–664, 2006.

[12] A. Auger. Convergence results for the (1,λ)-SA-ES using the theory of ϕ-irreducible Markov chains. *Theoretical Computer Science*, 334(1-3):35–69, 2005.

[13] A. Auger and N. Hansen. Linear convergence of comparison-based step-size adaptive randomized search via stability of Markov chains. Technical report, INRIA Saclay - Île de France, 2013.

[14] M. Baes. Estimate sequence methods: extensions and approximations. IFOR internal report, ETH Zürich, Switzerland, 2009.

[15] Y. Bao and R. Kan. On the moments of ratios of quadratic forms in normal random variables. *Journal of Multivariate Analysis*, 117(0):229–245, 2013.

[16] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[17] S. Becker, J. Bobin, and E. Cands. NESTA: A fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39, 2011.

[18] C. J. P. Bélisle, H. E. Romeijn, and R. L. Smith. Hit-and-Run algorithms for generating multivariate distributions. *Mathematics of Operations Research*, 18(2):255–266, 1993.

[19] E. Berndt, B. Hall, R. Hall, and J. A. Hausman. Estimation and inference in nonlinear structural models. *Annals of Economic and Social Measurement*, 3(4):103–116, 1974.

[20] A. Berny. Selection and reinforcement learning for combinatorial optimization. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Merelo, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VI*, volume 1917 of *LNCS*, pages 601–610. Springer Berlin Heidelberg, 2000.

[21] D. A. Berry and D. Stangl. *Bayesian biostatistics*. Dekker, Inc., New York, NY, USA, 1996.

[22] D. Bertsimas and S. Vempala. Solving convex programs by random walks. *Journal of the ACM*, 51(4):540–556, 2004.

[23] H.-G. Beyer. Toward a theory of evolution strategies: Some asymptotical results from the (1,+λ)-theory. *Evolutionary Computation, IEEE Transactions on*, 1(2):165–188, 1993.

[24] H.-G. Beyer. Towards a theory of "evolution strategies": Results for (1,+λ)-strategies on (nearly) arbitrary fitness functions. In Y. Davidor, H.-P. Schwefel, and R. Männer, editors, *Parallel Problem Solving from Nature - PPSN III*, volume 866 of *LNCS*, pages 57–67. Springer Berlin Heidelberg, 1994.

[25] H.-G. Beyer. *The theory of evolution strategies*. Natural Computing. Springer-Verlag, New York, NY, USA, 2001.

[26] H.-G. Beyer. Convergence analysis of evolutionary algorithms which are based on the paradigm of information geometry. *submitted*, 2013.

[27] H.-G. Beyer and H.-P. Schwefel. Evolution strategies a comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002.

[28] D. Bienstock and G. Iyengar. Solving fractional packing problems in O*(1/ϵ) iterations. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 146–155, New York, NY, USA, 2004. ACM.

[29] R. G. Bland, D. Goldfarb, and M. J. Todd. Feature article—the Ellipsoid method: A survey. *Operations Research*, 29(6):1039–1091, 1981.

[30] A. Boneh and A. Golan. Constraints' redundancy and feasible region boundedness by random feasible point generator (RFPG). In *3rd European Congress on Operations Research (EURO III), Amsterdam*, 1979.

[31] J. Bourgain. Random points in isotropic convex sets. In *Convex Geometric Analysis (Berkeley, CA, 1996). Math. Sci. Res. Inst. Publ.*, volume 34, pages 53–58. Cambridge University Press, Cambridge, 1999.

[32] J. Bourgain, J. Lindenstrauss, and V. Milman. Approximation of zonoids by zonotopes. *Acta Mathematica*, 162(1):73–141, 1989.

[33] G. E. Box and K. B. Wilson. On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 13(1):1–45, 1951.

[34] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004.

[35] C. Brif, R. Chakrabarti, and H. Rabitz. Control of quantum phenomena: past, present and future. *New Journal of Physics*, 12(7):075008, 2010.

[36] Y. Brise and B. Gärtner. Convergence rate of the DIRECT algorithm. Technical Report CGL-TR-47, ETH Zürich, 2012.

[37] S. H. Brooks. A discussion of random methods for seeking maxima. *Operations Research*, 6(2):244–251, 1958.

[38] C. G. Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90, 1970.

[39] W. J. Bühler. Two proofs of the Kantorovich inequality and some generalizations. *Revista colombiana de matematicas*, 21(1):147–154, 1987.

[40] M. A. Cauchy. Méthode générale pour la résolution des systèmes d'équations simultanées. *Comptes Rendus Hebd. Séances Acad. Sci.*, 25:536–538, 1847.

[41] T.-S. Chiang and Y. Chow. A limit theorem for a class of inhomogeneous Markov processes. *The Annals of Probability*, 17(4):1483–1502, 1989.

[42] P. Christiano, J. A. Kelner, A. Madry, D. A. Spielman, and S.-H. Teng. Electrical flows, Laplacian systems, and faster approximation of maximum flow in undirected graphs. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 273–282, New York, NY, USA, 2011. ACM.

[43] A. R. Conn, K. Scheinberg, and P. L. Toint. On the convergence of derivative-free methods for unconstrained optimization. In M. Buhmann and A. Iserles, editors, *Approximation Theory and Optimization, Tribute to M. J. D. Powell*. Cambridge University Press, Cambridge, 1996.

[44] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization*. MPS/SIAM Ser. Optim. SIAM, 2009.

[45] A. D. D. Craik. Prehistory of Faà di Bruno's formula. *The American Mathematical Monthly*, 112(2):119–130, 2005.

[46] D. Creal. A survey of sequential Monte Carlo methods for economics and finance. *Econometric Reviews*, 31(3):245–296, 2012.

[47] J. K. Cullum and R. A. Willoughby. *Lanczos Algorithms for Large Symmetric Eigenvalue Computations*. Society for Industrial and Applied Mathematics, 2002.

[48] W. C. Davidon. Variable metric method for minimization. *SIAM Journal on Optimization*, 1(1):1–17, 1991.

[49] K. R. Davidson and S. J. Szarek. Chapter 8: Local operator theory, random matrices and Banach spaces. In W. Johnson and J. Lindenstrauss, editors, *Handbook of the Geometry of Banach Spaces*, volume 1, pages 317–366. Elsevier Science Publishers Ltd., 2001.

[50] J. Dennis, Jr. and V. Torczon. Direct Search methods on parallel machines. *SIAM Journal on Optimization*, 1(4):448–474, 1991.

[51] F. Deutsch and H. Hundal. The rate of convergence for the method of Alternating Projections, II. *Journal of Mathematical Analysis and Applications*, 205(2):381–405, 1997.

[52] S. J. Dilworth, R. Howard, and J. W. Roberts. A general theory of almost convex functions. *Transactions of the AMS*, 358:3413–3445, 2006.

[53] J. C. Duchi., P. Bartlett, and M. Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012.

[54] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono. Optimal rates for zero-order optimization: the power of two function evaluations. Technical report, Stanford University, 2013.

[55] M. Dyer, A. Frieze, and R. Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *Journal of the ACM*, 38(1):1–17, 1991.

[56] Y. C. Eldar and D. Needell. Acceleration of randomized Kaczmarz method via the Johnson-Lindenstrauss lemma. *Numerical Algorithms*, 58(2):163–177, 2011.

[57] D. J. Evans. The use of pre-conditioning in iterative methods for solving linear equations with symmetric positive definite matrices. *IMA Journal of Applied Mathematics*, 4(3):295–314, 1968.

[58] H. G. Feichtinger, C. Cenker, M. Mayer, H. Steier, and T. Strohmer. New variants of the POCS method using affine subspaces of finite codimension with applications to irregular sampling. *Proc. SPIE: Visual Communications and Image Processing*, 1818:299–310, 1992.

[59] D. A. Flanders and G. Shortley. Numerical determination of fundamental modes. *Journal of Applied Physics*, 21(12):1326–1332, 1950.

[60] R. Fletcher. A new approach to variable metric algorithms. *The Computer Journal*, 13(3):317–322, 1970.

[61] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *The Computer Journal*, 7(2):149–154, 1964.

[62] C. Forbes, M. Evans, N. Hastings, and B. Peacock. *Statistical Distributions*. John Wiley & Sons, Inc., New York, NY, USA, 4 edition, 2011.

[63] O. Friedmann, T. D. Hansen, and U. Zwick. Subexponential lower bounds for randomized pivoting rules for the Simplex algorithm. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 283–292, New York, NY, USA, 2011. ACM.

[64] A. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 51(6):1025–1041, 2004.

[65] A. Galántai. On the rate of convergence of the alternating projection method in finite dimensional spaces. *Journal of Mathematical Analysis and Applications*, 310(1):30–44, 2005.

[66] B. Gärtner and V. Kaibel. Two new bounds for the Random-Edge Simplex algorithm. *SIAM Journal on Discrete Mathematics*, 21(1):178–190, 2007.

[67] M. K. Gavurin. The use of polynomials of best approximation for improving the convergence of iterative processes. *Uspekhi Matematicheskikh Nauk*, 5(3):156–160, 1950.

[68] G. Ghazal. Moments of the ratio of two dependent quadratic forms. *Statistics & Probability Letters*, 20(4):313–319, 1994.

[69] G. Ghazal. Recurrence formula for expectations of products of quadratic forms. *Statistics & Probability Letters*, 27(2):101–109, 1996.

[70] A. Giannopoulos, M. Hartzoulaki, and A. Tsolomitis. Random points in isotropic unconditional convex bodies. *Journal of the London Mathematical Society*, 72(3):779–798, 2005.

[71] A. Giannopoulos and V. Milman. Concentration property on probability spaces. *Advances in Mathematics*, 156(1):77–106, 2000.

[72] J. C. Gilbert, G. Le Vey, and J. Masse. La differentiation automatique de fonctions representees par des programmes. Rapport de recherche RR-1557, INRIA, 1991.

[73] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman Hall, New York, 1996.

[74] P. Gilmore and C. Kelley. An Implicit Filtering algorithm for optimization of functions with many local minima. *SIAM Journal on Optimization*, 5(2):269–285, 1995.

[75] T. Glasmachers, T. Schaul, S. Yi, D. Wierstra, and J. Schmidhuber. Exponential natural evolution strategies. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, pages 393–400, New York, NY, USA, 2010. ACM.

[76] F. Glover. Tabu Search–part I. *ORSA Journal on Computing*, 1(3):190–206, 1989.

[77] D. E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*, volume 412. Addison-Wesley Boston, 1989.

[78] D. Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of Computation*, 24(109):23–26, 1970.

[79] A. Goldstein. On Steepest Descent. *Journal of the Society for Industrial and Applied Mathematics Series A Control*, 3(1):147–151, 1965.

[80] G. H. Golub and C. F. V. Loan. *Matrix computations*. Johns Hopkins University Press, 2012.

[81] G. H. Golub and R. S. Varga. Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods. *Numerische Mathematik*, 3(1):157–168, 1961.

[82] J. G. D. Gooijer. Exact moments of the sample autocorrelations from series generated by general arima processes of order (p, d, q), $d = 0$ or 1. *Journal of Econometrics*, 14(3):365–379, 1980.

[83] R. Gordon, R. Bender, and G. T. Herman. Algebraic Reconstruction Techniques (ART) for three-dimensional electron microscopy and X-ray photography. *Journal of Theoretical Biology*, 29(3):471–481, 1970.

[84] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer Berlin Heidelberg, 1993.

[85] B. Grünbaum. Partitions of mass-distributions and of convex bodies by hyperplanes. *Pacific Journal of Mathematics*, 10(4):1257–1261, 1960.

[86] O. Guédon and M. Rudelson. l_p-moments of random vectors via majorizing measures. *Advances in Mathematics*, 208(2):798–823, 2007.

[87] H. Haario, E. Saksman, and J. Tamminen. Adaptive proposal distribution for random walk Metropolis algorithm. *Computational Statistics*, 14(3):375–395, 1999.

[88] H. Haario, E. Saksman, and J. Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242, 04 2001.

[89] N. Hansen. The CMA evolution strategy: A comparing review. In J. Lozano, P. Larraaga, I. Inza, and E. Bengoetxea, editors, *Towards a New Evolutionary Computation*, volume 192 of *Studies in Fuzziness and Soft Computing*, pages 75–102. Springer Berlin Heidelberg, 2006.

[90] N. Hansen. References to CMA-ES applications. available online, Dec. 2009. `https://www.lri.fr/~hansen/cmaapplications.pdf`.

[91] N. Hansen. The CMA evolution strategy: A tutorial. available online, 2011. `https://www.lri.fr/~hansen/cmatutorial.pdf`.

[92] N. Hansen, A. Auger, R. Ros, S. Finck, and P. Pošík. Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '10, pages 1689–1696, New York, NY, USA, 2010. ACM.

[93] N. Hansen, S. D. Muller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation, IEEE Transactions on*, 11(1):1–18, 2003.

[94] N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pages 312–317, 1996.

[95] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation, IEEE Transactions on*, 9(2):159–195, 2001.

[96] M. Hardt. The zen of Gradient Descent. blog, available online, Sept. 2013. `http://mrtz.org/blog/the-zen-of-gradient-descent/`.

[97] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[98] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 1994.

[99] E. Hazan. Sparse approximate solutions to semidefinite programs. In *Proceedings of the 8th Latin American conference on Theoretical informatics*, pages 306–316, Berlin, 2008. Springer-Verlag.

[100] R. Heijmans. When does the expectation of a ratio equal the ratio of expectations? *Statistical Papers*, 40:107–115, 1999.

[101] G. T. Herman. *Fundamentals of Computerized Tomography: Image Reconstruction from Projections*. Springer-Verlag, 1979.

[102] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, 1952.

[103] N. Higham. Optimization by Direct Search in matrix computations. *SIAM Journal on Matrix Analysis and Applications*, 14(2):317–333, 1993.

[104] N. J. Higham. *Functions of Matrices: Theory and Computation*. SIAM, 2008.

[105] B. Holmquist. Expectations of products of quadratic forms in normal variables. *Stochastic Analysis and Applications*, 14(2):149–164, 1996.

[106] R. Hooke and T. A. Jeeves. "Direct Search" solution of numerical and statistical problems. *Journal of the ACM*, 8(2):212–229, 1961.

[107] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, Cambridge, reprint 1990 edition, 1985.

[108] G. Hounsfield. Computerized transverse axial scanning (tomography): Part I. description of the system. *The British Journal of Radiology*, 46:1016–1022, 1973.

[109] T. Hu, V. Klee, and . Larman. Optimization of globally convex functions. *SIAM Journal on Control and Optimization*, 27(5):1026–1047, 1989.

[110] L. Isserlis. On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables. *Biometrika*, 12:134–139, 1918.

[111] J. Jägersküpper. Analysis of a simple evolutionary algorithm for minimization in Euclidean spaces. In J. C. Baeten, J. Lenstra, J. Parrow, and G. J. Woeginger, editors, *Automata, Languages and Programming*, volume 2719 of *LNCS*, pages 1068–1079. Springer Berlin Heidelberg, 2003.

[112] J. Jägersküpper. Rigorous runtime analysis of the (1+1) ES: 1/5-rule and ellipsoidal fitness landscapes. In *Foundations of Genetic Algorithms*, volume 3469 of *LNCS*, pages 356–361. Springer Berlin Heidelberg, 2005.

[113] J. Jägersküpper. How the (1+1) ES using isotropic mutations minimizes positive definite quadratic forms. *Theoretical Computer Science*, 361(1):38–56, 2006. Foundations of Genetic Algorithms Eighth Foundations of Genetic Algorithms Workshop 2005.

[114] J. Jägersküpper. Lower bounds for Hit-and-Run Direct Search. In J. Hromkovic, R. Královic, M. Nunkesser, and P. Widmayer, editors, *Stochastic Algorithms: Foundations and Applications*, volume 4665 of *LNCS*, pages 118–129. Springer Berlin Heidelberg, 2007.

[115] K. G. Jamieson, R. D. Nowak, and B. Recht. Query complexity of derivative-free optimization. In P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, NIPS, pages 2681–2689, 2012.

[116] M. Jebalia, A. Auger, and N. Hansen. Log-linear convergence and divergence of the scale-invariant (1+1)-ES in noisy environments. *Algorithmica*, 59(3):425–460, 2011.

[117] M. Jebalia, A. Auger, and P. Liardet. Log-linear convergence and optimal bounds for the (1+1)-ES. In N. Monmarché, E.-G. Talbi, P. Collet, M. Schoenauer, and E. Lutton, editors, *Artificial Evolution*, volume 4926 of *LNCS*, pages 207–218. Springer Berlin Heidelberg, 2008.

[118] D. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.

[119] D. Jones, C. Perttunen, and B. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

[120] D. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[121] M. Jones. On moments of ratios of quadratic forms in normal variables. *Statistics & Probability Letters*, 6(2):129–136, 1987.

[122] S. Kaczmarz. Angenäherte Auflösung von Systemen linearer Gleichungen. *Bulletin International de l'Academie Polonaise des Sciences et des Lettres*, 35:355–357, 1937.

[123] G. Kalai. A subexponential Randomized Simplex algorithm (extended abstract). In *Proceedings of the Twenty-fourth Annual ACM Symposium on Theory of Computing*, STOC '92, pages 475–482, New York, NY, USA, 1992. ACM.

[124] G. Kalai. Linear programming, the Simplex algorithm and simple polytopes. *Mathematical Programming*, 79(1-3):217–233, 1997.

[125] R. Kannan, L. Lovász, and M. Simonovits. Random walks and an O*(n^5) volume algorithm for convex bodies. *Random Structures & Algorithms*, 11(1):1–50, 1997.

[126] V. G. Karmanov. Convergence estimates for iterative minimization methods. *USSR Computational Mathematics and Mathematical Physics*, 14(1):1–13, 1974.

[127] V. G. Karmanov. On convergence of a random search method in convex minimization problems. *Theory of Probability and its applications*, 19(4):788–794, 1974. (in Russian).

[128] C. T. Kelley. *Iterative Methods for Optimization*. Number 18 in Frontiers in Applied Mathematics. SIAM Philadelphia, 1999.

[129] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948, 1995.

[130] J. Kiefer. Sequential minimax search for a maximum. *Proceedings of the AMS*, 4:502–506, 1953.

[131] K. V. Kim, Y. Nesterov, and B. V. Cherkasskiı. An estimate of the effort in computing the gradient. *Soviet Mathematics Doklady*, 29(2):384–387, 1984.

[132] S. Kirkpatrick, D. Gelatt, Jr., and M. P. Vecchi. Optimization by simmulated annealing. *Science*, 220(4598):671–680, 1983.

[133] G. Kjellström and L. Taxen. Stochastic optimization in system design. *Circuits Systems, IEEE Transactions on*, 28(7), 1981.

[134] A. Kleiner, A. Rahimi, and M. I. Jordan. Random Conic Pursuit for semidefinite programming. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, NIPS, pages 1135–1143. Curran Associates, Inc., 2010.

[135] J. N. Knight and M. Lunacek. Reducing the space-time complexity of the CMA-ES. In *Proceedings of the 9th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '07, pages 658–665. ACM, 2007.

[136] T. Kolda, R. Lewis, and V. Torczon. Optimization by Direct Search: New perspectives on some classical and modern methods. *SIAM Review*, 45(3):385–482, 2003.

[137] J. Kuczyński and H. Woźniakowski. Estimating the largest eigenvalue by the Power and Lanczos algorithms with a random start. *SIAM Journal on Matrix Analysis and Applications*, 13(4):1094–1122, 1992.

[138] A. Kumar. Expectation of product of quadratic forms. *Sankhyā: The Indian Journal of Statistics, Series B*, pages 359–362, 1973.

[139] C. Lanczos. Solution of systems of linear equations by minimized iterations. *Journal of Research of the National Bureau of Standards*, 49:33–53, 1952.

[140] Y. T. Lee and A. Sidford. Efficient accelerated Coordinate Descent methods and faster algorithms for solving linear systems. In *FOCS*, pages 147–156. IEEE Computer Society Press, 2013.

[141] D. Leventhal and A. Lewis. Randomized Hessian estimation and directional search. *Optimization*, 60(3):329–345, 2011.

[142] A. Y. Levin. On an algorithm for the minimization of convex functions (in russian). *Doklady Akademii nauk SSSR*, 160:1244–1247, 1965. (English translation: Soviet Mathematics Doklady, 6, 286–290, 1965.).

[143] R. M. Lewis, V. Torczon, and M. W. Trosset. Direct Search methods: then and now. *Journal of Computational and Applied Mathematics*, 124(1-2):191–207, 2000. Numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations.

[144] J. Lindeberg. Eine neue Herleitung des Exponentialgesetzes in der Wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 15(1):211–225, 1922.

[145] D. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, 1989.

[146] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer New York, 2008.

[147] M. Loog, J. J. Duistermaat, and L. M. J. Florack. On the behavior of spatial critical points under Gaussian blurring a folklore theorem and scale-space constraints. In M. Kerckhove, editor, *Scale-Space and Morphology in Computer Vision*, volume 2106 of *LNCS*, pages 183–192. Springer Berlin Heidelberg, 2001.

[148] I. Loshchilov. CMA-ES with restarts for solving CEC 2013 benchmark problems. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 369–376. IEEE Computer Society Press, 2013.

[149] I. Loshchilov, M. Schoenauer, and M. Sebag. BI-population CMA-ES algorithms with surrogate models and line searches. In *Genetic and Evolutionary Computation Conference (GECCO Companion)*, pages 1177–1184. ACM, 2013.

[150] L. Lovász and M. Simonovits. Random walks in a convex body and an improved volume algorithm. *Random Structures & Algorithms*, 4(4):359–412, 1993.

[151] L. Lovász and S. Vempala. Hit-and-Run from a corner. *SIAM Journal on Computing*, 35(4):985–1005, 2006.

[152] L. Lovász and S. Vempala. Simulated annealing in convex bodies and an volume algorithm. *Journal of Computer and System Sciences*, 72(2):392–417, 2006.

[153] J. R. Magnus. The moments of products of quadratic forms in normal variables. *Statistica Neerlandica*, 32(4):201–210, 1978.

[154] J. R. Magnus and B. Pesaran. Evaluation of moments of quadratic forms and ratios of quadratic forms in normal variables: background, motivation and examples. *Computational Statistics*, 8:21–37, 1993.

[155] K. Marti. Controlled random search procedures for global optimization. In V. Arkin, A. Shiraev, and R. Wets, editors, *Stochastic Optimization*, volume 81 of *Lecture Notes in Control and Information Sciences*, pages 457–474. Springer Berlin Heidelberg, 1986.

[156] A. M. Mathai and S. B. Provost. *Quadratic forms in random variables: theory and applications*. Number 126 in Statistics: textbooks and monographs. Dekker, Inc., New York, NY, USA, 1992.

[157] J. Matoušek. On variants of the Johnson-Lindenstrauss lemma. *Random Structures & Algorithms*, 33(2):142–156, 2008.

[158] J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16(4-5):498–516, 1996.

[159] J. Matyas. Random optimization. *Automation and Remote Control*, 26:246–253, 1965.

[160] S. Mendelson. On weakly bounded empirical processes. *Mathematische Annalen*, 340(2):293–314, 2008.

[161] S. Mendelson and A. Pajor. On singular values of matrices with independent rows. *Bernoulli*, 12(5):761–773, 2006.

[162] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[163] H. Mobahi and Y. Ma. Gaussian smoothing and asymptotic convexity. Technical report, University of Illinois at Urbana-Champaign, 2012.

[164] R. Morgan and M. Gallagher. Length scale for characterising continuous optimization problems. In C. A. C. Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia, and M. Pavone, editors, *Parallel Problem Solving from Nature - PPSN XII*, volume 7491 of *LNCS*, pages 407–416. Springer Berlin Heidelberg, 2012.

[165] E. Moulines and F. R. Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, NIPS, pages 451–459. Curran Associates, Inc., 2011.

[166] C. L. Müller. *Black-box Landscapes: Characterization, Optimization, Sampling, and Application to Geometric Configuration Problems*. PhD thesis, ETH Zürich, Switzerland, 2010.

[167] C. L. Müller and I. F. Sbalzarini. Gaussian adaptation revisited - an entropic view on covariance matrix adaptation. In C. Di Chio et al., editor, *EvoApplications*, number 6024 in LNCS, pages 432–441, Berlin, 2010. Springer-Verlag.

[168] C. L. Müller and I. F. Sbalzarini. Global characterization of the CEC 2005 fitness landscapes using fitness-distance analysis. In *Proceedings of the 2011 International Conference on Applications of Evolutionary Computation - Volume Part I*, EvoApplications'11, pages 294–303, Berlin, Heidelberg, 2011. Springer-Verlag.

[169] V. A. Mutseniyeks and L. A. Rastrigin. Extremal control of continuous multi-parameter systems by the method of random search. *Eng. Cybernetics*, 1:82–90, 1964.

[170] A. L. Nagar. The bias and moment matrix of the general k-class estimators of the parameters in simultaneous equations. *Econometrica*, 27(4):575–595, 1959.

[171] F. Natter. *The Mathematics of Computerized Tomography*. John Wiley & Sons, Inc., New York, NY, USA, 1986.

[172] D. Needell. Randomized Kaczmarz solver for noisy linear systems. *BIT Numerical Mathematics*, 50(2):395–403, 2010.

[173] D. Needell and R. Ward. Two-Subspace Projection method for coherent overdetermined systems. *Journal of Fourier Analysis and Applications*, 19(2):256–269, 2013.

[174] N. S. Needell, Deanna and R. Ward. Stochastic Gradient Descent and the randomized Kaczmarz algorithm. *arXiv:1310.5715v3*, 2013.

[175] J. A. Nelder and R. Mead. A Simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[176] A. S. Nemirovski. Efficient methods in convex programming. available online, 1995. `http://www2.isye.gatech.edu/~nemirovs/Lect_EMCO.pdf`.

[177] A. S. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

[178] A. S. Nemirovski and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. John Wiley & Sons, Inc., New York, NY, USA, 1983.

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

[180] Y. Nesterov. On an approach to the construction of optimal methods of minimization of smooth convex functions. *Ekonomika i Matematicheskie Metody*, 24:509–517, 1988.

[181] Y. Nesterov. *Introductory Lectures on Convex Optimization*. Kluwer Academic Publishers, Boston, USA, 2004.

[182] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[183] Y. Nesterov. Rounding of convex sets and efficient gradient methods for linear programming problems. *Optimization Methods Software*, 23(1):109–128, 2008.

[184] Y. Nesterov. Random gradient-free minimization of convex functions. Technical report, ECORE, 2011.

[185] Y. Nesterov. Efficiency of Coordinate Descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[186] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[187] Y. Nesterov and A. S. Nemirovskii. *Interior Point Polynomial Algorithms in Convex Programming*, volume 13 of *Studies in Applied Mathematics*. SIAM, 1994.

[188] H. Neudecker. The Kronecker matrix product and some of its applications in econometrics. *Statistica Neerlandica*, 22(1):69–82, 1968.

[189] D. J. Newman. Location of the maximum on unimodal surfaces. *Journal of the ACM*, 12(3):395–398, 1965.

[190] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer New York, 2 edition, 2006.

[191] Y. Ollivier, L. Arnold, A. Auger, and N. Hansen. Information-geometric optimization algorithms: A unifying picture via invariance principles. Technical report, INRIA Saclay - Île de France, 2013.

[192] J. E. Orosz and S. H. Jacobson. Finite-time performance analysis of static simulated annealing algorithms. *Computational Optimization and Applications*, 21(1):21–53, 2002.

[193] Z. Páles. On approximately convex functions. *Proceedings of the AMS*, 131:243–252, 2003.

[194] M. S. Paolella. Computing moments of ratios of quadratic forms in normal variables. *Computational Statistics & Data Analysis*, 42(3):313–331, 2003.

[195] G. Paouris. Concentration of mass on convex bodies. *Geometric & Functional Analysis GAFA*, 16(5):1021–1049, 2006.

[196] E. J. G. Pitman. The "closest" estimates of statistical parameters. *Mathematical Proceedings of the Cambridge Philosophical Society*, 33:212–222, 1937.

[197] B. Polyak. *Introduction to Optimization*. Optimization Software, Inc., New York, NY, USA, 1987.

[198] J. Ponstein. Seven kinds of convexity. *SIAM Review*, 9(1):115–119, 1967.

[199] M. Powell. A Direct Search optimization method that models the objective and constraint functions by linear interpolation. In S. Gomez and J.-P. Hennart, editors, *Advances in Optimization and Numerical Analysis*, volume 275 of *Mathematics and Its Applications*, pages 51–67. Springer Netherlands, 1994.

[200] M. Powell. UOBYQA: unconstrained optimization by quadratic approximation. *Mathematical Programming*, 92(3):555–582, 2002.

[201] M. Powell. The NEWUOA software for unconstrained optimization without derivatives. In G. Pillo and M. Roma, editors, *Large-Scale Nonlinear Optimization*, volume 83 of *Nonconvex Optimization and Its Applications*, pages 255–297. Springer US, 2006.

[202] M. J. D. Powell. Direct search algorithms for optimization calculations. *Acta Numerica*, 7:287–336, 1998.

[203] V. Protasov. Algorithms for approximate calculation of the minimum of a convex function from its values. *Mathematical Notes*, 59(1):69–74, 1996.

[204] S. Puntanen, G. P. H. Styan, and J. Isotalo. *Matrix Tricks for Linear Statistical Models: Our Personal Top Twenty*. Springer Berlin Heidelberg, 2011.

[205] L. A. Rademacher. Approximating the centroid is hard. In *Proceedings of the Twenty-third Annual Symposium on Computational Geometry*, SCG '07, pages 302–305, New York, NY, USA, 2007. ACM.

[206] L. B. Rail. *Automatic Differentiation: Techniques and Applications*, volume 120 of *LNCS*. Springer-Verlag, 1981.

[207] L. A. Rastrigin. The convergence of the random search method in the extremal control of a many-parameter system. *Automation and Remote Control*, 24:1337–1342, 1963.

[208] L. Rastrygin. Problems of random search. *Radiophysics and Quantum Electronics*, 15(7):747–754, 1972.

[209] I. Rechenberg. *Evolutionsstrategie; Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, 1973.

[210] L. Rios and N. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293, 2013.

[211] F. Romeo and A. Sangiovanni-Vincentelli. A theoretical framework for simulated annealing. *Algorithmica*, 6(1–6):302–345, 1991.

[212] H. H. Rosenbrock. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3):175–184, 1960.

[213] M. Rudelson. Random vectors in the isotropic position. *Journal of Functional Analysis*, 164(1):60–72, 1999.

[214] M. Rudelson and R. Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM*, 54(4):21, 2007.

[215] Y. Saad and H. A. van der Vorst. Iterative solution of linear systems in the 20th century. *Journal of Computational and Applied Mathematics*, 123(12):1–33, 2000. Numerical Analysis 2000. Vol. III: Linear Algebra.

[216] T. Sawa. Finite-sample properties of the k-class estimators. *Econometrica*, 40(4):653–680, 1972.

[217] T. Sawa. The exact moments of the least squares estimator for the autoregressive model. *Journal of Econometrics*, 8(2):159–172, 1978.

[218] C. Schaffer. A conservation law for generalization performance. In W. W. Cohen and H. Hirsch, editors, *Proceedings of the Eleventh International Machine Learning Conference*, pages 259–265. Rutgers University, New Brunswick, NJ, 1994.

[219] T. Schaul. Natural evolution strategies converge on sphere functions. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, GECCO '12, pages 329–336, New York, NY, USA, 2012. ACM.

[220] M. Schumer and K. Steiglitz. Adaptive step size random search. *Automatic Control, IEEE Transactions on*, 13(3):270–276, 1968.

[221] H.-P. Schwefel. *Evolutionsstrategie und numerische Optimierung*. PhD thesis, Technische Universität Berlin, 1975.

[222] H.-P. Schwefel. *Evolution and Optimum Seeking: The Sixth Generation*. John Wiley & Sons, Inc., New York, NY, USA, 1993.

[223] M. I. Sezan and H. Stark. *Image Recovery: Theory and Application*, chapter Applications of convex projection theory to image recovery in tomography and related areas, pages 155–270. Academic Press, Inc., 1987.

[224] D. F. Shanno. Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation*, 24(111):647–656, 1970.

[225] N. Shor. Cut-off method with space extension in convex programming problems. *Cybernetics*, 13(1):94–96, 1977.

[226] G. Shortley. Use of Tschebyscheff-polynomial operators in the numerical solution of boundary-value problems. *Journal of Applied Physics*, 24(4):392–396, 1953.

[227] B. Shubert. A sequential method seeking the global maximum of a function. *SIAM Journal on Numerical Analysis*, 9(3):379–388, 1972.

[228] V. Simoncini and D. B. Szyld. Recent computational developments in Krylov subspace methods for linear systems. *Numerical Linear Algebra with Applications*, 14(1):1–59, 2007.

[229] M. D. Smith. On the expectation of a ratio of quadratic forms in normal variables. *Journal of Multivariate Analysis*, 31(2):244–257, 1989.

[230] R. L. Smith. Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research*, 32(6):1296–1308, 1984.

[231] W. Spendley, G. R. Hext, and F. R. Himsworth. Sequential application of simplex designs in optimisation and evolutionary operation. *Technometrics*, 4(4):441–461, 1962.

[232] G. W. Stewart, III. A modification of Davidon's minimization method to accept difference approximations of derivatives. *Journal of the ACM*, 14(1):72–83, 1967.

[233] S. U. Stich. On low complexity acceleration techniques for randomized optimization. In T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith, editors, *Parallel Problem Solving from Nature - PPSN XIII*, volume 8672 of *LNCS*, pages 130–140. Springer International Publishing, 2014.

[234] S. U. Stich and B. Gärtner. Random Pursuit in Hilbert space. Technical report, ETH Zürich, 2013. Technical Report CGL-TR-88.

[235] S. U. Stich and C. L. Müller. On spectral invariance of randomized Hessian and covariance matrix adaptation schemes. In C. A. C. Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia, and M. Pavone, editors, *Parallel Problem Solving from Nature - PPSN XII*, volume 7491 of *LNCS*, pages 448–457. Springer Berlin Heidelberg, 2012.

[236] S. U. Stich, C. L. Müller, and B. Gärtner. Supporting online material for optimization of convex functions with Random Pursuit. *arXiv:1111.0194v2*, 2012.

[237] S. U. Stich, C. L. Müller, and B. Gärtner. Variable metric Random Pursuit. *submitted, arXiv:1210.5114v3*, 2012.

[238] S. U. Stich, C. L. Müller, and B. Gärtner. Matrix-valued iterative random projections. Technical report, ETH Zürich, 2013. Technical Report CGL-TR-87.

[239] S. U. Stich, C. L. Müller, and B. Gärtner. Optimization of convex functions with Random Pursuit. *SIAM Journal on Optimization*, 23(2):1284–1309, 2013.

[240] T. Strohmer and R. Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 15(2):262–278, 2009.

[241] J. Sun, J. M. Garibaldi, and C. Hodgman. Parameter estimation using metaheuristics in systems biology: A comprehensive review. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 9(1):185–202, 2012.

[242] K. Tanabe. Projection method for solving a singular system of linear equations and its applications. *Numerische Mathematik*, 17(3):203–214, 1971.

[243] V. Torczon. On the convergence of Pattern Search algorithms. *SIAM Journal on Optimization*, 7(1):1–25, 1997.

[244] G. A. Tribello, M. Ceriotti, and M. Parrinello. A self-learning algorithm for biased molecular dynamics. *Proceedings of the National Academy of Sciences*, 107(41):17509–17514, 2010.

[245] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal on Optimization*, 2008.

[246] P. Vaidya. A new algorithm for minimizing convex functions over convex sets. *Mathematical Programming*, 73(3):291–341, 1996.

[247] D. Vanderbilt and S. G. Louie. A Monte Carlo simulated annealing approach to optimization over continuous variables. *Journal of Computational Physics*, 56(2):259–271, 1984.

[248] S. Vempala. Recent progress and open problems in algorithmic convex geometry. In K. Lodaya and M. Mahajan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 8 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 42–64, Dagstuhl, Germany, 2010.

[249] R. Vershynin. Approximating the moments of marginals of high-dimensional distributions. *The Annals of Probability*, 39(4):1591–1606, 2011.

[250] J. H. M. Wedderburn. *Lectures on Matrices (Colloquium Publications)*. American Mathematical Society, New York, 1938.

[251] T. Whitney and R. Meany. Two algorithms related to the method of Steepest Descent. *SIAM Journal on Numerical Analysis*, 4(1):109–118, 1967.

[252] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber. Natural evolution strategies. *Journal of Machine Learning Research*, 15:949–980, 2014.

[253] D. Wierstra, T. Schaul, J. Peters, and J. Schmidhuber. Natural evolution strategies. In *Evolutionary Computation, 2008. CEC 2008. IEEE Congress on*, pages 3381–3387, 2008.

[254] P. Wolfe. Convergence conditions for ascent methods. *SIAM Review*, 11(2):226–235, 1969.

[255] P. Wolfe. Convergence conditions for ascent methods. II: Some corrections. *SIAM Review*, 13(2):185–188, 1971.

[256] D. Wolpert and W. Macready. No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1):67–82, 1997.

[257] D. H. Wolpert and W. G. Macready. No free lunch theorems for search. Working papers, Santa Fe Institute, 1995.

[258] D. J. Woods. *An Iteractive Approach for Solving Multi-objetive Optimization Problems*. PhD thesis, Rice University, Houston, Texas, USA, 1979.

[259] M. H. Wright. Direct search methods: Once scorned, now respectable. In *Pitman Research Notes in Mathematics Series*, pages 191–208, 1995.

[260] S. Yi, D. Wierstra, T. Schaul, and J. Schmidhuber. Stochastic search using the Natural Gradient. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1161–1168, New York, NY, USA, 2009. ACM.

[261] T. Ypma. Historical development of the Newton-Raphson method. *SIAM Review*, 37(4):531–551, 1995.

[262] D. B. Yudin and A. S. Nemirovskii. Evaluation of the informational complexity of mathematical programming problems. *Ekonomika i Matematicheskie Metody*, 12:128–142, 1976.

[263] D. B. Yudin and A. S. Nemirovskii. Informational complexity and effective methods of solution for convex extremal problems. *Ekonomika i Matematicheskie Metody*, 12:357–369, 1976.

[264] R. Zieliński and P. Neumann. *Stochastische Verfahren zur Suche nach dem Minimum einer Funktion.* Akademie-Verlag, Berlin, Germany, 1983.

[265] A. Zouzias and N. M. Freris. Randomized extended Kaczmarz for solving least squares. *SIAM Journal on Matrix Analysis Applications*, 34(2):773–793, 2013.

Index