Université catholique de Louvain

UCLouvain
Institute for Information
and Communication Technologies,
Electronics and Applied Mathematics

CORE
CENTER FOR
OPERATIONS RESEARCH
AND ECONOMETRICS

Quasi-Newton Methods with Provable Efficiency Guarantees

Anton Rodomanov

Thesis submitted in partial fulfillment
of the requirements for the degree of
Docteur en Sciences de l'Ingénieur

Dissertation committee:

Prof. Yurii Nesterov (Université catholique de Louvain, Advisor)
Prof. Pierre-Antoine Absil (Université catholique de Louvain)
Prof. Volkan Cevher (École Polytechnique Fédérale de Lausanne)
Prof. François Glineur (Université catholique de Louvain)
Prof. Michael L. Overton (New York University)

August, 2022

To my dear wife Julia and daughter Alisa

Abstract

Quasi-Newton methods are very popular in Optimization. They have a long, rich history, and perform extremely well for solving real-life problems. However, almost nothing is known about theoretical *efficiency guarantees* for these methods.

The goal of this work is the advancement of the theory of quasi-Newton methods. This includes both obtaining new *convergence estimates* for the already existing algorithms and developing new methods with provable *efficiency guarantees*.

In this thesis, we present our results in several directions. First, we provide a new theoretical analysis of local superlinear convergence of classical quasi-Newton methods and establish *explicit* and *non-asymptotic* bounds on their rate of convergence. Then, we develop and analyze new quasi-Newton methods which have some advanced features. Specifically, we propose a new family of *greedy* quasi-Newton methods for which, apart from local superlinear convergence, it is also possible to guarantee the convergence of Hessian approximations. Finally, we study one algorithm which is related to classical quasi-Newton methods, namely, the Ellipsoid Method, and develop a new variant of this method, which has a better dependency on the dimensionality of the problem than the standard one.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my advisor, Professor Yurii Nesterov, for his guidance, for all our meetings and fruitful discussions, for giving me the freedom to explore various research topics and always being open to new ideas and suggestions, for being a constant source of inspiration. It has been a real pleasure to work with him, and I am very grateful for his kindness and continuous support.

I am also extremely grateful to the members of my dissertation committee, Professors Pierre-Antoine Absil, Volkan Cevher, François Glineur, and Michael L. Overton, for their valuable time and efforts spent on reviewing this thesis. Their feedback was very useful and greatly helped to improve the quality of the presentation.

Many thanks to my friends and colleagues, Nikita Doikov, Geovani Nunes Grapiglia, Radu-Alexandru Dragomir, Alexander Gasnikov, Pavel Dvurechensky, Evgeniia Vorontsova, Masoud Ahookhosh, Mihai Florea, and Valentin Leplat, for numerous discussions and the time we spent together.

Special thanks to the administrative staff at UCLouvain, in particular, to Marie-Christine Joveneau, Pascale Premereur, Margaux Hubin, Nancy Guillaume, Marie Gonze, Etienne Huens, and Catherine Germain, for always being there ready to help.

I would also like to extend my sincere gratitude to my bachelor's and master's advisors, Dmitry Vetrov and Dmitry Kropotov, who introduced me to science and who taught me a lot. Without them, I would not be writing this thesis.

Last but not least, I am extremely thankful to my wife Julia for her constant support and understanding.

This thesis has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 788368).

Contents

Chapter 1

Introduction

1.1 Motivation

Optimization methods are among the most important numerical algorithms in Computational Mathematics. They are used for solving the problems of minimizing or maximizing a certain function subject to certain constraints. Such problems often arise in different applications from Machine Learning, Statistics, Economics, Transport Modeling, Telecommunications, Signal Processing, etc. [14].

One of the simplest optimization methods is the Gradient Method, which can be traced back to Cauchy [31]. Each iteration of this method requires computing only the gradient of the objective function and is usually relatively cheap. However, the convergence of the Gradient Method may be very slow if the problem is ill-conditioned, which is often the case in practice.

For ill-conditioned problems, a much more efficient algorithm is Newton's Method. It has a very fast quadratic convergence and is insensitive to the conditioning of the problem. However, compared to the Gradient Method, each iteration of Newton's Method is much more expensive: in addition to the gradient, one also needs to compute the Hessian matrix and solve a linear system with this matrix.

A natural idea is to combine the Gradient and Newton methods together. This leads to quasi-Newton methods, which have a reputation of the most efficient numerical schemes for solving large-scale optimization problems. Quasi-Newton methods can be seen as an approximation of the standard Newton's Method, in which the exact Hessian is replaced by some matrix,

which is updated in iterations according to certain low-rank formulas involving the gradients of the objective function.

There exist numerous quasi-Newton methods that differ mainly in the rules of updating Hessian approximations. The three most popular variants are the DFP [39, 62], BFGS [18, 19, 59, 70, 169] and SR1 [17, 39] methods. From the computational experience point of view, BFGS is typically regarded as the most efficient quasi-Newton scheme [144].

Unfortunately, from the theoretical point of view, very little is known about convergence guarantees for quasi-Newton methods. Despite a large amount of research in the area, the main theoretical results about quasi-Newton methods are still the old results which were obtained several decades ago. These results state, in one form or another, that, under suitable assumptions, certain quasi-Newton methods eventually converge to a solution and that the rate of convergence is locally superlinear: the ratio of successive residuals tends to zero when the number of iterations goes to infinity.

However, there are no concrete *efficiency estimates* or *complexity bounds* for quasi-Newton methods. The main problem is that the aforementioned theoretical results are all *asymptotic* and *nonexplicit* in nature: they do not provide us with any particular inequalities which can be used to estimate the number of iterations required to achieve a certain accuracy. Note that we cannot really blame the authors for this since, in that time, the majority of convergence results in Optimization were indeed asymptotic. On the contrary, it was a great achievement of that time when the corresponding works on convergence results for quasi-Newton methods first appeared.

Nevertheless, since then, Optimization Theory has advanced a lot. There has been a shift in the paradigm. A central place in Optimization Theory is now occupied by Complexity Theory, which originated from the work of Yudin and Nemirovski [193]. In this theory, it is not enough to simply have results of the form that "the method eventually converges" or "a certain limit is zero". Instead, for each method, it is customary to obtain explicit efficiency estimates, discuss how these estimates depend on the parameters of the *problem class*, compare different methods with each other not only in terms of the type of convergence (linear, superlinear, etc.) but also in terms of the complexity bounds, etc.

Strangely enough, during all these years, the theory of quasi-Newton methods has not caught up with the shift in the paradigm. It seems as if most of the recent research in the area was concerned more with generalizing or adapting existing methods and ideas to different problem formulations and settings.

Thus, there is a clear need in developing and modernizing the theory of quasi-Newton methods. This includes, in particular, obtaining explicit efficiency estimates, investigating lower complexity bounds, finding optimal methods, etc. Note that, in general, this is a vast and difficult topic, so we can only hope to cover part of it. In this thesis, our focus will be on *local* efficiency estimates (meaning that the method starts with a sufficiently good initial point).

In addition to the classical quasi-Newton methods, discussed above, there exist other methods which are closely related. The most famous of them is probably the Ellipsoid Method. In this method, there is also some scaling matrix which is updated at each iteration according to a certain low-rank formula involving the (sub)gradient of the objective function. In this regard, the Ellipsoid Method can also be considered a quasi-Newton method, even though it was not originally constructed with the goal of approximating Newton's Method.

As opposed to classical quasi-Newton methods, the Ellipsoid Method has always had a solid theoretical foundation. In fact, it was developed by its authors exactly with the purpose of constructing an implementable method with the complexity bound close to the optimal one. Therefore, there has never been a question about explicit efficiency estimates for the Ellipsoid Method.

Nevertheless, there are still some issues with the Ellipsoid Method that have not been solved up to now. One of these issues, which we address in this thesis, is that the Ellipsoid Method does not withstand the passage to the limit when the dimension of the space goes to infinity: it stops converging at all. As a result, the method has poor performance on problems of large dimension and, in particular, it can even be slower than the basic (Sub)Gradient Method.

Our interest in studying the Ellipsoid Method is twofold. On the one hand, as was already mentioned, it is a method of quasi-Newton type, for which there already exist some efficiency estimates. Therefore, the ideas and tools used in the analysis of the Ellipsoid Method could be helpful for developing the theory of classical quasi-Newton methods and constructing new algorithms with theoretical guarantees. On the other hand, all improvements in the Ellipsoid Method are valuable in themselves as they could lead to more efficient approaches for solving various applied problems.

The main goal of this thesis is the advancement of the theory of quasi-Newton methods. This includes the analysis of already existing methods as well as the development of new methods with *provable efficiency guarantees*.

1.2 Historical Overview

In Section 1.2.1, we review the history of quasi-Newton methods and some recent trends. In Section 1.2.2, we review the main advancements in the theory of Optimization (mostly related to Complexity Theory) which have been made since the invention of quasi-Newton methods. In Section 1.2.3, we review the history of the Ellipsoid Method and its role in Optimization.

1.2.1 Quasi-Newton Methods

The main references on quasi-Newton methods are arguably the monographs by Nocedal and Wright [144], Fletcher [61], and Dennis and Schnabel [47], as well as the 1977 paper by Dennis and Moré [46]. The historical development of quasi-Newton methods and, in particular, BFGS, is covered in detail in the recent survey by Papakonstantinou [145].

The first quasi-Newton method for minimizing nonlinear functions was proposed by Davidon in 1959 [39] under the name of a "variable metric method". Davidon's method was further studied and improved by Fletcher and Powell in 1963 [62]. This method has been known as the *Davidon–Fletcher–Powell (DFP)* Method since then.

For the problem of solving a system of nonlinear equations, first quasi-Newton methods were developed by Broyden in 1965 [16]. He proposed two new algorithms for approximating the Jacobian matrix which are now commonly known as the *Good Broyden* and *Bad Broyden* methods. The main feature of these methods is that they produce Jacobian approximations which are, in general, not symmetric.

An important step in the history of quasi-Newton methods was made in 1970 by Broyden [18, 19], Fletcher [59], Goldfarb [70] and Shanno [169] who discovered the celebrated *Broyden–Fletcher–Goldfarb–Shanno (BFGS)* Method. It is very interesting that the four authors independently arrived at the same formula from different considerations.

In the same year, Powell [148] proposed a special technique for symmetrizing the Good Broyden update. He obtained a new quasi-Newton algorithm for optimization problems, which has subsequently been referred to as the *Powell Symmetric Broyden (PSB)* Method. Powell's technique was later used by Dennis in 1971 [42] to show that almost all well-known quasi-Newton methods could be derived using the same approach.

The *Symmetric Rank-One (SR1)* formula was first presented by Davidon in the appendix of his 1959 report [39]. However, it was later rediscovered

several times by different authors.

An interesting idea that quasi-Newton updates can be derived from variational considerations using the *least change* principle was first formulated by Greenstadt in 1970 [81]. It was exactly this idea that Goldfarb used to arrive at the BFGS formula [70]. In 1977, Dennis and Moré [46] demonstrated that many other popular quasi-Newton updates could also be derived by using the same principle.

First *convergence guarantees* for quasi-Newton methods for nonlinear problems were obtained by Powell in 1971 [149]. He considered the DFP Method with *exact line search*, applied to minimizing a strongly convex function, and showed that this method has global convergence and that asymptotically the rate of convergence is superlinear. A year later, Dixon [49, 50] extended this result to BFGS, SR1 and many other methods by establishing a remarkable result that all quasi-Newton algorithms from Broyden's family [17] coincide under exact line search.

In 1973, Broyden, Dennis and Moré [20] showed that, for the purposes of *local convergence*, there is no need to apply any line search in quasi-Newton methods, as one can simply use *unit step sizes*, similarly to Newton's Method. For DFP, BFGS and several other quasi-Newton methods, they proved superlinear convergence under the assumption that both the starting point and the initial Hessian (or Jacobian) approximation are sufficiently good. The analysis was based on the "bounded deterioration" principle, introduced earlier by Dennis [43, 44].

The superlinear convergence proofs obtained by Powell and by Broyden, Dennis and Moré were later unified by Dennis and Moré in 1974 [45]. They established a necessary and sufficient condition for superlinear convergence of quasi-Newton methods. They also showed that, for DFP and BFGS, the superlinear convergence automatically follows from the assumption that the sequence of residuals is summable, provided that the method eventually switches to unit step sizes. This result was further extended to the entire convex Broyden class by Griewank and Toint in 1982 [83].

In 1976, Powell [150] proved that BFGS with an *inexact line search*, based on Wolfe conditions, has *global convergence* on strongly convex functions. His analysis was based on studying the evolution of the trace and determinant of Hessian approximation matrices. He also showed, applying the tools of Dennis and Moré, that asymptotically the rate of convergence is superlinear, provided that the method always chooses the unit step size when it is admissible by the line search. Powell's results were later extended by Byrd, Nocedal and Yuan [25] to all methods from the convex Broyden

class excluding DFP. In 1989, Byrd and Nocedal [24] simplified Powell's original analysis by introducing a special potential function which combines the trace and the logarithm of determinant. They also made an interesting observation that the new potential function provides an alternative way for proving superlinear convergence, compared to the standard Dennis–Moré approach.

Another line of research is related to convergence of Hessian approximation matrices constructed by quasi-Newton methods. One impressive result, proved by Ge and Powell in 1983 [66], is that the matrices, produced by BFGS and DFP methods, are actually convergent, even though their limit may be different from the true Hessian of the objective function. Later, Stoer [175] generalized this result onto other methods from the convex Broyden class. In 1991, Conn, Gould and Toint [34] showed that, whenever a certain uniform linear independence assumption is satisfied, the SR1 Method generates Hessian approximations that do converge to the true Hessian. A similar assumption was used by Boggs and Tolle in 1994 [11] in their analysis of Broyden's class.

In order to apply quasi-Newton methods for solving large-scale problems and overcome the need for storing the Hessian approximation matrices in memory, a number of approaches were proposed, known as *limited-memory quasi-Newton methods*. These methods originated with the works of Perry [147] and Shanno [170] and are based on the idea of combining quasi-Newton and nonlinear conjugate gradient methods together. Currently, the most popular algorithm of this type is the L-BFGS Method, presented by Liu and Nocedal in 1989 [113].

For large-scale problems possessing a certain *structure*, there exist other techniques for reducing memory requirements of quasi-Newton methods. One of the most important examples is given by *sparse problems*. The research on sparse quasi-Newton methods started with the work of Schubert in 1970 [167] who proposed a version of Broyden's update for solving a sparse system of nonlinear equations. Symmetric quasi-Newton updates, taking into account the sparsity pattern of the Hessian, were later developed by Toint [184–188], Shanno [171], and Fletcher [60]. A slightly different approach was taken by Griewank and Toint [84] who introduced partitioned quasi-Newton methods for partially separable problems [82].

The growing popularity of quasi-Newton methods for unconstrained optimization quickly lead to the extension of these algorithms to the problems of *constrained optimization* under the frameworks of Sequential Quadratic Programming (SQP) and Augmented Lagrangian methods. The first steps

in this direction were made in the late 1970s by Garcia-Palomares and Mangasarian [64], Han [87–89], Powell [151, 152] and Tapia [177]. A subsequent study and further development of these ideas was carried out by Coleman and Conn [33], Nocedal and Overton [143], Tapia [178], and Byrd, Tapia and Zhang [27], among many others.

Although quasi-Newton methods were originally designed for smooth optimization, they also turn out to be quite efficient for *nonsmooth problems*, even without any special modifications. This phenomenon was first studied by Lewis and Overton [109] and later by Guo [86]. Nevertheless, in general, there are almost no theoretical guarantees, and there are indeed examples of nonsmooth problems for which classical quasi-Newton methods do fail. To address this issue, various approaches have been proposed [35–37, 192].

First extensions of quasi-Newton methods to *Riemannian optimization* were developed by Gabay already in 1982 [63]. However, a significant interest in such methods has emerged only recently due to several important applications. A number of new algorithms have been proposed in the last decade [15, 90–92, 154, 163]. For a general introduction to optimization on manifolds, see excellent monographs by Absil et al. [1] and Boumal [13].

In recent years, there has also been a large amount of research on optimization methods for the problems of *stochastic* and *finite-sum optimization*, which often arise in Machine Learning. In these problems, the direct computation of the gradient is an expensive operation and should be avoided as much as possible. Many quasi-Newton methods have been proposed for this kind of problems [26, 71, 73, 116, 117, 119, 166, 174, 190].

1.2.2 Evolution of Optimization Theory

After the invention of quasi-Newton methods, Optimization Theory has advanced a lot. Let us briefly review some major achievements in this area which are mainly related to Complexity Theory and have served as a motivation for our research.

Complexity Theory for numerical optimization methods was developed by Nemirovski and Yudin in their famous monograph in 1979 [126]. Although it was not widely recognized at first, eventually, it has had a major effect on the field of Optimization. By introducing the notion of an *oracle*, Nemirovski and Yudin formalized the concept of an *optimization method* and its *complexity* for a given *class of problems*. For many general classes of optimization problems, they obtained *lower complexity bounds* and found *optimal methods*. One of the most important results of this work was jus-

tification of the fact that *convex* optimization problems are the ones which admit efficient algorithms, in contrast to nonconvex ones, which are, in general, unsolvable (at least with provable guarantees of efficiency).

A major event which happened in Optimization in the 1980s and 1990s was the so-called *Interior-Point Revolution*. It started in 1984 when Karmarkar [95] invented the first interior-point method. Specifically, Karmarkar developed a new polynomial-time algorithm for solving Linear Programming (LP) problems. At that time, there already existed one polynomial algorithm for LP by Khachiyan [97], which was based on the Ellipsoid Method. However, in practice, this algorithm was no match for the Simplex Method, which had been the main algorithm for solving LP back then. Even though it was known that the Simplex Method could work exponentially slowly in certain theoretical scenarios, in real practice, it always worked much better than was expected in the worst case. However, the situation with Karmarkar's algorithm was completely different: not only was it efficient in theory, but it was also competitive with the Simplex Method in practice. Naturally, Karmarkar's discovery sparked a lot of interest in interior-point methods among many researchers. An important step was made by Renegar in 1988 [156] when he presented the first path-following interior-point method for LP, which was based on the classical scheme of *logarithmic barrier methods*. Later, Nesterov and Nemirovski discovered which properties of the logarithmic barrier for LP were responsible for efficiency of the associated interior-point methods. More importantly, they showed that, for many other problems, there exist easily computable barriers with the same properties. They called such barriers *self-concordant*. The work of Nesterov and Nemirovski, carried out in the late 1980s and reflected in their 1994 monograph [139], significantly extended the scope of interior-point methods to many classes of convex nonlinear optimization problems. This work forms the foundation of the modern theory of interior-point methods.

Starting from the 2000s, many optimization problems, arising in various applications, became too big and out of reach of powerful interior-point methods. The only methods which could be applied for solving these problems were gradient methods with simple iterations. Naturally, the focus of the optimization community changed to studying and improving such methods. A lot of progress has been made since then, especially in connection with Structural Optimization.

The first milestone was the invention of *Smoothing Technique* by Nesterov in 2005 [128]. He noticed that many nonsmooth problems from real-

world applications possess a certain saddle structure, which could be explicitly used for efficiently approximating such problems with smooth ones. Applying the Fast Gradient Method [127] to the corresponding smooth approximation, one obtains a method for solving the original problem whose complexity is much better than that of standard black-box subgradient methods. A closely related discovery, based on completely different considerations, was made slightly later by Nemirovski [123] when he proposed the Mirror-Prox Method for solving smooth saddle-point problems. This method has almost the same scope of application and the same complexity as Nesterov's Smoothing Technique.

Another major advancement was the discovery of efficient methods for the problems of *Composite Optimization*. In these problems, the objective function is the sum of two components: a smooth one, given by a black-box oracle, and a general convex function with simple structure (e.g., the indicator of a set or a certain regularizer). In 2013, Nesterov [132] showed that, despite the absence of good properties of the sum, such problems can be efficiently solved by special gradient methods with efficiency typical for the smooth part of the objective.

Significant progress has been made in the development of methods for solving the problems of *Stochastic Optimization* and its particular but very important case of *Finite-Sum Optimization*. A modern comprehensive treatment of the Stochastic Gradient Method, including its complexity analysis, was given by Nemirovski et al. in 2009 [124]. Computation of accuracy certificates for this method was studied by Lan, Nemirovski and Shapiro in 2012 [103]. In the same year, Lan [101] presented an accelerated version of the Stochastic Gradient Method, which was later improved by Ghadimi and Lan [67, 68]. *Adaptive* stochastic gradient methods, which are provably more efficient than their nonadaptive counterparts in certain favorable situations, were proposed by Duchi, Hazan and Singer in 2011 [56], by Kingma and Ba in 2014 [98], by Levy, Yurtsever and Cevher in 2018 [108], by Kavis et al. in 2019 [96], and by Alacaoglu et al. in 2020 [3], among others. An important group of algorithms, especially suited for Finite-Sum Optimization, is *variance-reduced methods*. These methods originated with the works of Le Roux, Schmidt and Bach (2012) [105, 164], and Johnson and Zhang (2013) [93], and are currently among the best methods for minimizing large sums of functions. Lower complexity bounds for Finite-Sum Optimization were established by Lan and Zhou [104], by Arjevani and Shamir [5], and by Woodworth and Srebro [191]. For more details on stochastic optimization and variance-reduced methods, including their

history, see the monograph by Lan [102] and the recent survey by Gower et al. [75].

Special attention should be given to *randomized coordinate descent methods*. The first complexity analysis of these methods for general functions was carried out by Nesterov in 2012 [131]. He demonstrated that randomized coordinate descent algorithms possess good efficiency estimates and can outperform standard gradient methods in a variety of applications. He also presented an accelerated version of randomized coordinate descent with uniform sampling for unconstrained problems. This method was later improved and generalized to composite problems by Lu and Xiao [114], by Fercoq and Richtárik [58], and by Lin, Lu and Xiao [111]. It was also shown, by Lee and Sidford in 2013 [106], by Allen-Zhu et al. [4] in 2016, and by Nesterov and Stich in 2017 [141], that, for unconstrained problems, further acceleration could be achieved by using nonuniform sampling strategies. An important contribution of Lee and Sidford's work [106], among other things, is that they proposed a special technique which can be used for getting rid of full-dimensional operations, arising in standard implementations of accelerated coordinate descent. The first extension of coordinate descent to composite optimization problems was described by Richtárik and Takáč in 2014 [157] for the basic non-accelerated method. Primal-dual coordinate descent algorithms were developed by Shalev-Shwartz and Zhang in 2013 [168], by Lin, Lu and Xiao in 2015 [112], by Qu et al. in 2016 [155], by Tran-Dinh, Fercoq and Cevher in 2018 [189], and by Alacaoglu, Fercoq and Cevher in 2020 [2]. An interesting variant of coordinate descent, which can solve linearly constrained convex problems over networks, was proposed by Necoara, Nesterov and Glineur in 2017 [120]. The first nonuniform strategy of coordinate sampling, suitable for composite problems, was developed by Perekrestenko, Cevher and Jaggi [146].

While a large part of the optimization community has been interested in first-order methods, some major advancements have been made for *second-order methods*. First global efficiency estimates for second-order methods were obtained by Nesterov and Polyak in 2006 [140] for the cubic regularization of Newton's Method (also known as the Cubic Newton Method). An adaptive variant of this algorithm was developed by Cartis, Gould and Toint in 2011 [28, 29]. The first accelerated version of the Cubic Newton Method was proposed by Nesterov in 2008 [129]. Another accelerated version with a better iteration complexity bound was described by Monteiro and Svaiter in 2013 [118]. Universal variants, automatically adapting to the actual level of smoothness of the objective function, were first proposed by Grapiglia

and Nesterov in 2017 [76] and then further accelerated and generalized to composite optimization problems by the same authors in 2019 [77]. An interesting result, justifying the superiority of the Cubic Newton Method over the Gradient Method on the class of strongly convex functions with Lipschitz continuous gradient, was established by Doikov and Nesterov in 2021 [54] (see also the discussion at the end of Section 2.4.2).

Tensor methods are based on the natural idea of further accelerating second-order methods by using higher-order derivatives. These methods have been known in Optimization for a long time (see, e.g., [12, 165]). The first complexity analysis of tensor methods for convex problems was carried out by Baes in 2009 [7]. For nonconvex problems, these methods were studied, among others, by Birgin et al. in 2017 [9], by Martínez in 2017 [115] and by Cartis et al. in 2019 [30]. However, up until recently, the practical applicability of tensor methods was really questionable due to the hardness of minimizing nonconvex multivariate polynomials. This situation was changed after the seminal paper by Nesterov [135]. Specifically, he showed that the auxiliary problems arising in tensor methods for minimizing *convex* functions are themselves convex (for an appropriate choice of the regularization parameter). Furthermore, he demonstrated that third-order methods can be efficiently implemented at virtually the same cost as second-order methods [135, 138]. This discovery sparked significant interest in tensor methods for convex optimization. Inexact versions of these methods were proposed by Grapiglia and Nesterov [78, 80], by Nesterov [134], and by Doikov and Nesterov [52]. Accelerated variants with nearly optimal complexities were studied by Gasnikov et al. in 2019 [65]. Universal versions were developed by Grapiglia and Nesterov in 2020 [79]. Local convergence of tensor methods for composite optimization problems was studied by Doikov and Nesterov in 2021 [53].

Recently, there has been a growing interest in *computer-assisted analysis* of optimization methods. This approach originated from the work of Drori and Teboulle in 2014 [55] and relies on semidefinite programming performance estimation framework. For more details, see the works by Taylor, Hendrickx and Glineur [180–183] and by De Klerk, Glineur and Taylor [41].

Another interesting research direction is the study of the influence of *inexact information* on the performance of optimization methods. This topic was extensively studied, among others, by d'Aspremont in 2008 [38], by Dvurechensky and Gasnikov in 2016 [57], by Necoara, Patrascu and Glineur in 2019 [121], by Kamzolov, Dvurechensky and Gasnikov in 2020 [94], and by Stonyakin et al. in 2021 [176]. Special attention is deserved by the 2014

work of Devolder, Glineur and Nesterov [48], where the authors introduced the notion of inexact first-order oracle and provided many interesting examples.

1.2.3 Ellipsoid Method

The Ellipsoid Method was invented by Yudin and Nemirovski in 1976 [193] for solving general convex programming problems. Their original motivation was related to Complexity Theory. After studying lower complexity bounds for black-box optimization methods, they came to the conclusion that the Center-of-Gravity Method, proposed by Levin [107] and Newman [142] in 1965, was optimal in terms of the oracle calls. However, each iteration of that method required computing the center of gravity of a convex polytope—an operation whose arithmetical complexity is potentially exponential in the dimension. In an attempt to make the Center-of-Gravity Method practical, Yudin and Nemirovski proposed a modified version of this algorithm, in which the polytopes were replaced by ellipsoids. They showed that the oracle complexity of the resulting method was close to the optimal one.

Later, it turned out that the Ellipsoid Method was a particular instance of the general scheme of *subgradient methods with space dilation*, introduced by Shor in 1970 [172]. This point was clarified by Shor in his 1977 paper [173], where he presented the Ellipsoid Method in a simpler form, similar to that of quasi-Newton methods.

In 1979, Khachiyan [97] used the Ellipsoid Method for proving his famous result on *polynomial solvability* of Linear Programming. More precisely, he indicated how the standard Ellipsoid Method could be modified for checking the feasibility of a system of linear inequalities with integer data in polynomial time. This was a major event in Theoretical Computer Science, which sparked a lot of interest in the Ellipsoid Method (for more details, see [10]). Inspired by Khachiyan's result, in 1981, Grötschel, Lovász and Schrijver [85] applied the Ellipsoid Method for establishing polynomial solvability of a number of important problems arising in Combinatorial Optimization.

Soon after the invention of the Ellipsoid Method, it became clear that this method could also be applied for solving *general problems with convex structure*, such as saddle-point problems, Nash equilibrium problems, variational inequalities, etc. However, for a long time, there was one difficulty with this approach. Specifically, the procedure for generating approximate

solutions to such problems required solving an auxiliary piecewise linear optimization problem (see, e.g., Sections 5 and 6 in [122]). Although this auxiliary computation did not use any additional calls to the oracle, it was still computationally expensive and, in some cases, could take even more time than the Ellipsoid Method itself. In 2010, Nemirovski, Onn and Rothblum [125] successfully resolved this difficulty by proposing an efficient technique for computing approximate solutions.

For more details and historical remarks on the Ellipsoid Method, see the monographs by Nemirovski [122] and by Ben-Tal and Nemirovski [8]; also see an excellent survey by Bland, Goldfarb and Todd [10].

1.3 Contributions of this Thesis

Let us briefly summarize the main contributions of this thesis.

Superlinear Rates for Classical Quasi-Newton Methods

In Chapter 3, we study the local convergence of classical quasi-Newton methods for nonlinear optimization. Although it was well established a long time ago that asymptotically these methods converge superlinearly, the corresponding rates of convergence were still remaining unknown. We address this problem. We obtain first explicit non-asymptotic rates of superlinear convergence for the standard quasi-Newton methods, which are based on the updating formulas from the convex Broyden class. In particular, this class includes the famous DFP and BFGS methods. The main parameters in the corresponding efficiency estimates are the dimension of the problem and its condition number.

The contents of this chapter is based on the following articles [160, 161]:

- A. Rodomanov and Y. Nesterov. New Results on Superlinear Convergence of Classical Quasi-Newton Methods. *Journal of Optimization Theory and Applications*, 188:744–769, 2021.

- A. Rodomanov and Y. Nesterov. Rates of superlinear convergence for classical quasi-Newton methods. *Mathematical Programming*, 194:159–190, 2022.

Greedy Quasi-Newton Methods

In Chapter 4, we study greedy variants of quasi-Newton methods. They are based on the updating formulas from a certain subclass of the Broyden family. In particular, this subclass includes the well-known DFP, BFGS, and SR1 updates. However, in contrast to the classical quasi-Newton methods, which use the difference of successive iterates for updating the Hessian approximations, our methods apply basis vectors, greedily selected so as to maximize a certain measure of progress. For greedy quasi-Newton methods, we establish an explicit non-asymptotic bound on their rate of local superlinear convergence, as applied to minimizing strongly convex functions with Lipschitz continuous gradient and Hessian. The established superlinear convergence rate contains a contraction factor which depends on the square of the iteration counter. We also show that greedy quasi-Newton methods produce Hessian approximations whose deviation from the exact Hessians linearly converges to zero.

The contents of this chapter is based on the following article [159]:

- A. Rodomanov and Y. Nesterov. Greedy Quasi-Newton Methods with Explicit Superlinear Convergence. *SIAM Journal on Optimization*, 31(1):785–811, 2021.

Subgradient Ellipsoid Method

In Chapter 5, we present a new ellipsoid-type algorithm for solving nonsmooth problems with convex structure. Examples of such problems include nonsmooth convex minimization problems, convex-concave saddle-point problems and variational inequalities with monotone operator. The new algorithm can be seen as a combination of the standard Subgradient and Ellipsoid methods. However, in contrast to the latter one, the proposed method has a reasonable convergence rate even when the dimensionality of the problem is large. For generating accuracy certificates in our algorithm, we propose an efficient technique which improves upon the previously known one [125].

The contents of this chapter is based on the following article [162]:

- A. Rodomanov and Y. Nesterov. Subgradient ellipsoid method for nonsmooth convex problems. *Mathematical Programming*, 2022.

1.4 Overview of Main Results

Let us provide a quick overview of the main results presented in this thesis.

First, in Chapter 3, we obtain explicit and nonasymptotic rates of *local* convergence for classical quasi-Newton methods from the convex Broyden class, as applied to unconstrained minimization of a smooth function. The function class which we consider is strongly convex functions with Lipschitz continuous gradient and Hessian. The following table displays our results for the two most important members of the convex Broyden class—BFGS and DFP methods.

Classical quasi-Newton methods (Algorithm 3.4.1)

Method	τ_k	Convergence rate **(new)**	Starting moment **(new)**	Convergence to Hessian
BFGS	0	$[\varkappa^{n/k} - 1]^{k/2}\sqrt{\varkappa}$	$n \ln \varkappa$	No
DFP	1	$[\varkappa(\varkappa^{n/k} - 1)]^{k/2}\sqrt{\varkappa}$	$n\varkappa \ln \varkappa$	No

In this table, k is the iteration counter; $\varkappa$ is the condition number of the objective function; n is the dimension of the space; the convergence rate is presented in terms of the inaccuracy measure λ_k/λ_0, where λ_k is the local norm of the gradient, defined in (3.4.5); "Starting moment" is the starting moment of superlinear convergence; "Convergence to Hessian" reflects whether the Hessian approximations constructed by the method are guaranteed to converge to the exact Hessian. For the sake of simplicity, we have also omitted several absolute constants (for more precise formulas, see Theorem 3.4.8 and the accompanying discussion).

According to our results, BFGS has a much better convergence rate than DFP. In particular, its starting moment of superlinear convergence is almost insensitive to the condition number $\varkappa$ (it is under the logarithm). On the other hand, we see that classical quasi-Newton methods do not guarantee the convergence to the exact Hessian.

We address the latter problem in Chapter 4. Specifically, we show that we can replace the classical rule for selecting the update direction in standard quasi-Newton methods with a new *greedy* rule, and obtain new *greedy quasi-Newton methods* that do guarantee the convergence of Hessian approximations to the exact Hessian. These new methods also have another interesting element—a special correction procedure—ensuring that the Hessian approximations constructed by the method are actually *upper* approx-

imations. As a result, it becomes possible to work with a subclass of the Broyden family, which is larger than the classical convex Broyden class and includes, in particular, not only BFGS and DFP updates, but also SR1. We summarize the convergence properties of greedy quasi-Newton methods in the table below, where we use the same notation as before and omit all absolute constants (for more precise formulas, see Theorem 4.3.8 and Section 4.4); χ_k^{BFGS} is the special value[1] corresponding to the BFGS update.

New greedy quasi-Newton methods (Algorithm 4.3.1)

Method	χ_k	Convergence rate	Starting moment	Convergence to Hessian
GrSR1 GrDFP GrBFGS	0 1 χ_k^{BFGS}	$\exp\bigl(-k^2/(n\varkappa)\bigr)(n\varkappa)^k$	$n\varkappa \ln(n\varkappa)$	Yes

As we can see, for greedy quasi-Newton methods, we have a worse starting moment of superlinear convergence than for the classical methods. However, the rate of convergence of greedy methods is asymptotically faster, and their Hessian approximations are more accurate. For a more detailed comparison of greedy methods with the classical ones, see Section 4.4.

In the final part of this thesis (Chapter 5), we develop a new Subgradient Ellipsoid Method (Algorithm 5.2.1) for solving general nonsmooth problems with convex structure. The new method can be seen as an attempt to improve the classical Ellipsoid Method by correcting its behavior in the case when the dimension of the space is large. Below we present the *global* convergence rate of the Subgradient Ellipsoid Method (in terms of a special "gap" measure δ_k defined in Section 5.1.2) and compare it with those of the standard Subgradient and Ellipsoid methods.

Algorithms for nonsmooth convex problems

Algorithm	Convergence rate	Withstands limiting passage $n \to \infty$
Subgradient Method	$1/\sqrt{k}$	Yes
Ellipsoid Method	$\exp(-k/n^2)$	No
Subgr. Ellipsoid Method **(new)**	$\min^*\{1/\sqrt{k}, \exp(-k/n^2)\}$	Yes

[1] Specifically, $\chi_k^{\text{BFGS}} := \langle \nabla^2 f(x_{k+1})u_k, u_k\rangle / \langle \tilde{G}_k u_k, u_k\rangle$, see (4.1.5).

In the above table, k is the iteration counter; n is the dimension of the space; for simplicity, all absolute constants (and R) are omitted; $\min^*\{a_k, b_k\}$ is a certain expression which coincides with $\min\{a_k, b_k\}$ for all values of k except a "small" interval between n^2 and $n^2 \ln n$ (see Section 5.3.4 for more details). The conclusion is that the rate of the Subgradient Ellipsoid Method is virtually the best among those of the standard Subgradient and Ellipsoid methods.

Chapter 2

Background

In this chapter, we first introduce our general notation and review some important definitions and facts which we will use throughout this thesis. We then review several standard optimization methods and discuss their convergence guarantees. Altogether, the results presented in this chapter serve as a foundation for our future developments.

The contents of this chapter is mostly based on several classical monographs such as [8, 61, 102, 122, 133, 144]. For the reader's convenience, we also present all accompanying proofs unless they are too long or too technical.

2.1 Notation and Generalities

2.1.1 Vector Spaces

Everywhere in this thesis, if not stated explicitly, we denote by $\mathbb{E}$ an arbitrary *finite-dimensional real vector space*. Its *dual space*, composed of all linear functionals on $\mathbb{E}$, is denoted by $\mathbb{E}^*$. Note that these spaces are of the same dimension: $\dim \mathbb{E} = \dim \mathbb{E}^*$. The value of a linear functional $s \in \mathbb{E}^*$, evaluated at a point $x \in \mathbb{E}$, is denoted by $\langle s, x \rangle$. The form $\langle \cdot, \cdot \rangle$, defined in this way, is bilinear and nondegenerate, and is called the (canonical) *dual pairing* between $\mathbb{E}$ and $\mathbb{E}^*$.

Of course, the most important example is $\mathbb{E} = \mathbb{R}^n$. In this case, we usually make an identification $\mathbb{E}^* = \mathbb{R}^n$ in such a way that the dual pairing $\langle \cdot, \cdot \rangle$

corresponds to the standard dot product:

$$\langle s, x \rangle = \sum_{i=1}^{n} s_i x_i, \qquad \forall s, x \in \mathbb{R}^n, \tag{2.1.1}$$

where s_i and x_i are the coordinates of s and x, respectively.

In principle, no generality would be lost if we assumed that $\mathbb{E} = \mathbb{E}^* = \mathbb{R}^n$, as we can always choose a basis in $\mathbb{E}$ and identify every element of $\mathbb{E}$ with its coordinate representation. However, in general, it is better to work more abstractly without fixing any particular coordinate system and by keeping distinction between $\mathbb{E}$ and its dual $\mathbb{E}^*$, as this approach less obscures the fundamental geometric and algebraic structure underlying the main objects we work with.

Note that, in contrast to $\mathbb{E}^*$, the *double dual space* $\mathbb{E}^{**}$, resulting by taking the dual of $\mathbb{E}^*$, can always be naturally identified with $\mathbb{E}$ (without fixing any coordinate representations):

$$\mathbb{E}^{**} = \mathbb{E}, \tag{2.1.2}$$

in the sense that the transformation $\tilde{x} \colon \mathbb{E} \to \mathbb{E}^{**}$, that maps each $x \in \mathbb{E}$ to $\tilde{x} \equiv \tilde{x}(x) \in \mathbb{E}^{**}$, defined by $\langle \tilde{x}, s \rangle = \langle s, x \rangle$ for all $s \in \mathbb{E}^*$, is a bijection.

Sometimes, we work with two or more finite-dimensional real vector spaces at the same time. In this case, unless explicitly mentioned, we usually denote them by $\mathbb{E}_1$, $\mathbb{E}_2$, etc.

The vector space of all linear operators from $\mathbb{E}_1$ to $\mathbb{E}_2$ is denoted by

$$\mathcal{L}(\mathbb{E}_1, \mathbb{E}_2) \coloneqq \{A \colon \mathbb{E}_1 \to \mathbb{E}_2 \mid A \text{ is a linear operator}\}.$$

In this notation, $\mathbb{E}^* = \mathcal{L}(\mathbb{E}, \mathbb{R})$.

When $\mathbb{E}_1 = \mathbb{R}^n$ and $\mathbb{E}_2 = \mathbb{R}^m$, each linear operator $A \colon \mathbb{E}_1 \to \mathbb{E}_2$ is usually identified with a real $m \times n$ matrix, and $\mathcal{L}(\mathbb{E}_1, \mathbb{E}_2)$ corresponds to the space $\mathbb{R}^{m \times n}$ of real $m \times n$ matrices.

For more details about abstract finite-dimensional vector spaces, their properties and duality, we refer the reader to [99].

2.1.2 Adjoint Operator

For a linear operator $A\colon \mathbb{E}_1 \to \mathbb{E}_2^*$, its *adjoint*[1] $A^*\colon \mathbb{E}_2 \to \mathbb{E}_1^*$ is the unique linear operator satisfying

$$\langle Ax_1, x_2\rangle = \langle A^* x_2, x_1\rangle, \qquad \forall x_1 \in \mathbb{E}_1,\ \forall x_2 \in \mathbb{E}_2. \tag{2.1.3}$$

A linear operator $A\colon \mathbb{E} \to \mathbb{E}^*$ is called *self-adjoint*[2] if $A = A^*$. The space of all self-adjoint linear operators from $\mathbb{E}$ to $\mathbb{E}^*$ is denoted by

$$\mathcal{S}(\mathbb{E}, \mathbb{E}^*) \coloneqq \{A \in \mathcal{L}(\mathbb{E}, \mathbb{E}^*) : A = A^*\}.$$

Note that $\mathcal{S}(\mathbb{E}, \mathbb{E}^*)$ is a linear subspace of $\mathcal{L}(\mathbb{E}, \mathbb{E}^*)$.

In the special case when $\mathbb{E}_1 = \mathbb{E}_1^* = \mathbb{R}^n$ and $\mathbb{E}_2 = \mathbb{E}_2^* = \mathbb{R}^m$ (with the standard identification of operators with matrices), the adjoint is the matrix transpose: $A^* = A^T$. Similarly, when $\mathbb{E} = \mathbb{E}^* = \mathbb{R}^n$, each self-adjoint linear operator $A\colon \mathbb{E} \to \mathbb{E}^*$ is just a symmetric matrix, and $\mathcal{S}(\mathbb{E}, \mathbb{E}^*)$ is the space $\mathbb{S}^n$ of real symmetric $n \times n$ matrices.

Sometimes, in the formulas involving products of linear operators, it is convenient to treat $s \in \mathbb{E}^*$ as the linear operator $s\colon \mathbb{R} \to \mathbb{E}^*$ defined by

$$s\alpha \coloneqq \alpha s, \qquad \forall \alpha \in \mathbb{R}.$$

Under the standard identification $\mathbb{R}^* = \mathbb{R}$, the adjoint of this operator is

[1] Recall that the spaces $\mathbb{E}_1$ and $\mathbb{E}_2$ are allowed to be arbitrary. Therefore, this definition is actually more general than it may seem at a first glance. In particular, combined with (2.1.2), it allows us to naturally define the adjoint for any operator not only from $\mathcal{L}(\mathbb{E}_1, \mathbb{E}_2^*)$, but also from $\mathcal{L}(\mathbb{E}_1^*, \mathbb{E}_2)$, $\mathcal{L}(\mathbb{E}_1, \mathbb{E}_2)$ and $\mathcal{L}(\mathbb{E}_1^*, \mathbb{E}_2^*)$. For example, to define the adjoint of an operator $H\colon \mathbb{E}_1^* \to \mathbb{E}_2$, we can consider the spaces $\mathbb{E}_1' \coloneqq \mathbb{E}_1^*$ and $\mathbb{E}_2' \coloneqq \mathbb{E}_2^*$. Then, H can be naturally identified with $\tilde{H}\colon \mathbb{E}_1' \to (\mathbb{E}_2')^*$ defined by $\tilde{H}s_1 \coloneqq \tilde{x}_2(Hs_1)$, where $\tilde{x}_2\colon \mathbb{E}_2 \to \mathbb{E}_2^{**}$ is the canonical isomorphism between $\mathbb{E}_2$ and $\mathbb{E}_2^{**}$. For such an operator, (2.1.3) provides us with the definition of the adjoint $\tilde{H}^*\colon \mathbb{E}_2' \to (\mathbb{E}_1')^*$, or more explicitly, $\tilde{H}^*\colon \mathbb{E}_2^* \to \mathbb{E}_1^{**}$. This operator, in turn, can be naturally identified with $H^*\colon \mathbb{E}_2^* \to \mathbb{E}_1$ defined by $H^* s_2 \coloneqq \tilde{x}_1^{-1}(\tilde{H}^* s_2)$, where $\tilde{x}_1\colon \mathbb{E}_1 \to \mathbb{E}_1^{**}$ is the canonical isomorphism between $\mathbb{E}_1$ and $\mathbb{E}_1^{**}$. In the end, we obtain, for any $s_1 \in \mathbb{E}_1^*$ and $s_2 \in \mathbb{E}_2^*$,

$$\langle s_2, Hs_1\rangle = \langle \tilde{x}_2(Hs_1), s_2\rangle = \langle \tilde{H}s_1, s_2\rangle = \langle \tilde{H}^* s_2, s_1\rangle = \langle s_1, \tilde{x}_1^{-1}(\tilde{H}^* s_2)\rangle = \langle s_1, H^* s_2\rangle.$$

In other words, the adjoint of an operator $H\colon \mathbb{E}_1^* \to \mathbb{E}_2$, is the operator $H^*\colon \mathbb{E}_2^* \to \mathbb{E}_1$ defined by $\langle s_2, Hs_1\rangle = \langle s_1, H^* s_2\rangle$ for all $s_1 \in \mathbb{E}_1$ and $s_2 \in \mathbb{E}_2$. Similarly, one can show that the adjoints of operators $U\colon \mathbb{E}_1 \to \mathbb{E}_2$ and $T\colon \mathbb{E}_1^* \to \mathbb{E}_2^*$ are, respectively, the operators $U^*\colon \mathbb{E}_2^* \to \mathbb{E}_1^*$ and $T^*\colon \mathbb{E}_2 \to \mathbb{E}_1$ defined by $\langle s_2, Ux_1\rangle = \langle U^* s_2, x_1\rangle$ and $\langle Ts_1, x_2\rangle = \langle s_1, T^* x_2\rangle$ for any $s_1 \in \mathbb{E}_1^*$, $s_2 \in \mathbb{E}_2^*$, $x_1 \in \mathbb{E}_1$ and $x_2 \in \mathbb{E}_2$.

[2] As in Footnote 1, replacing $\mathbb{E}$ with $\mathbb{E}^*$ and using (2.1.2), we obtain the natural definition of self-adjointness for an operator $H\colon \mathbb{E}^* \to \mathbb{E}$. A similar remark applies to many other subsequent definitions in the current and the following sections.

then $s^*\colon \mathbb{E} \to \mathbb{R}$ defined by

$$s^* x \coloneqq \langle s, x \rangle, \qquad \forall x \in \mathbb{E}.$$

In particular, for any $s_1 \in \mathbb{E}_1^*$ and $s_2 \in \mathbb{E}_2^*$, the product $s_1 s_2^*$ can be treated as the rank-one linear operator from $\mathcal{L}(\mathbb{E}_1, \mathbb{E}_2^*)$:

$$(s_2 s_1^*) x_1 = \langle s_1, x_1 \rangle s_2, \qquad \forall x_1 \in \mathbb{E}_1.$$

For us, the following result will be especially useful, as it provides an explicit formula for the inverse operator of a rank-one perturbation.

Proposition 2.1.1 (Sherman–Morrison formula)**.** *Let $A\colon \mathbb{E}_1 \to \mathbb{E}_2^*$ be a nondegenerate linear operator and let $s_1 \in \mathbb{E}_1^*$, $s_2 \in \mathbb{E}_2^*$. Then,*

$$(A + s_2 s_1^*)^{-1} = A^{-1} - \frac{A^{-1} s_2 s_1^* A^{-1}}{1 + \langle s_1, A^{-1} s_2 \rangle}, \tag{2.1.4}$$

provided that $1 + \langle s_1, A^{-1} s_2 \rangle \neq 0$.

Proof. This formula can be easily verified by multiplying the right-hand side of (2.1.4) by $A + s_2 s_1^*$ and checking that the result is the identity operator. □

2.1.3 Order Between Self-Adjoint Operators

The *partial order* for any $A_1, A_2 \in \mathcal{S}(\mathbb{E}, \mathbb{E}^*)$ is defined in the standard way:

$$A_1 \preceq A_2 \quad \iff \quad \langle A_1 x, x \rangle \leq \langle A_2 x, x \rangle, \quad \forall x \in \mathbb{E}. \tag{2.1.5}$$

Similarly, we can define the *strict order*:

$$A_1 \prec A_2 \quad \iff \quad \langle A_1 x, x \rangle < \langle A_2 x, x \rangle, \quad \forall x \in \mathbb{E} \setminus \{0\}.$$

An operator $A \in \mathcal{S}(\mathbb{E}, \mathbb{E}^*)$ is called *positive semidefinite* if $A \succeq 0$, and *positive definite* if $A \succ 0$. The cone of all self-adjoint positive semidefinite linear operators from $\mathbb{E}$ to $\mathbb{E}^*$ is denoted by

$$\mathcal{S}_+(\mathbb{E}, \mathbb{E}^*) \coloneqq \{A \in \mathcal{S}(\mathbb{E}, \mathbb{E}^*) : A \succeq 0\}.$$

The interior of this cone is formed by positive definite operators:

$$\mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*) \coloneqq \{A \in \mathcal{S}(\mathbb{E}, \mathbb{E}^*) : A \succ 0\}.$$

In the case when $\mathbb{E} = \mathbb{E}^* = \mathbb{R}^n$ (with the standard identification of operators with matrices), $\mathcal{S}_+(\mathbb{E}, \mathbb{E}^*)$ and $\mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ are the spaces $\mathbb{S}^n_+$ and $\mathbb{S}^n_{++}$ of symmetric positive semidefinite and symmetric positive definite real $n \times n$ matrices, respectively.

2.1.4 Derivatives

Let $T\colon Q \to \mathbb{E}_2$ be a mapping, defined on a set $Q \subseteq \mathbb{E}$ with $\operatorname{int} Q \neq \varnothing$. Then, T is called differentiable at a point $x \in \operatorname{int} Q$ if there exists an operator $DT(x) \in \mathcal{L}(\mathbb{E}, \mathbb{E}_2)$, called the *derivative* of T at x, such that[3]

$$T(x+h) = T(x) + DT(x)h + o(\|h\|) \tag{2.1.6}$$

for all sufficiently small $h \in \mathbb{E}$, where $\|\cdot\|$ is an arbitrary norm in $\mathbb{E}$. Note that the derivative does not depend on the particular choice of the norm, as all norms in a finite-dimensional space generate the same topology. Given a set $Q' \subseteq \operatorname{int} Q$, we call T differentiable on Q' if T is differentiable at every point from Q'.

The derivative DT is itself a mapping from a certain set $Q' \subseteq \operatorname{int} Q \subseteq \mathbb{E}$, on which T is differentiable, to a certain vector space, namely, $\mathcal{L}(\mathbb{E}, \mathbb{E}_2)$. If $\operatorname{int} Q' \neq \varnothing$ and DT is differentiable at a point $x \in \operatorname{int} Q'$, we say that f is *twice differentiable* at x, and define the *second derivative* of T at x, denoted by $D^2T(x)$, as the derivative of DT at x. Given a set $Q'' \subseteq \operatorname{int} Q'$, we call T twice differentiable on Q'' if T is twice differentiable at any point from Q''. Similarly, we can define the notions of differentiability and derivative of order 3, 4, etc.

Thus, for each point $x \in \operatorname{int} Q$, at which T is differentiable as many times as necessary, the derivatives of T at x are certain linear operators of the following form:

$$\begin{aligned} DT(x) &\in \mathcal{L}(\mathbb{E}, \mathbb{E}_2) =: \mathcal{L}_1, \\ D^2T(x) &\in \mathcal{L}(\mathbb{E}, \mathcal{L}_1) =: \mathcal{L}_2, \\ D^3T(x) &\in \mathcal{L}(\mathbb{E}, \mathcal{L}_2) =: \mathcal{L}_3, \\ &\cdots . \end{aligned} \tag{2.1.7}$$

If T is $p \geq 1$ times differentiable at $x \in \operatorname{int} Q$, then, for any integer $1 \leq k \leq p$

[3] Here we use the standard notation $o(\|h\|)$ to denote an arbitrary element $\Delta(h) \in \mathbb{E}_2$ satisfying $\|\Delta(h)\|/\|h\| \to 0$ as $h \to 0$. In other words, (2.1.6) is equivalent to the following statement: $\|T(x+h) - T(x) - DT(x)h\|/\|h\| \to 0$ as $h \to 0$.

and any $h_1, \dots, h_k \in \mathbb{E}$, we denote

$$D^p T(x)[h_1, \dots, h_k] \coloneqq D^p T(x) h_1 \dots h_k. \tag{2.1.8}$$

The mapping $D^p T(x)[\cdot, \dots, \cdot]$, defined in this way, is a *symmetric multilinear* operator from $\mathbb{E}^k$ to $\mathcal{L}_{p-k}$ (with $\mathcal{L}_0 \coloneqq \mathbb{E}_2$). When all arguments in (2.1.8) are the same, i.e., $h_1 = \dots = h_k \equiv h$, we use the abbreviation $D^p T(x)[h]^k$.

When dealing with a *functional*, i.e., a mapping of the form $f \colon Q \to \mathbb{R}$, where $Q \subseteq \mathbb{E}$ is a set with $\operatorname{int} Q \neq \varnothing$, we usually prefer to call the first two derivatives Df and $D^2 f$ (provided, of course, they exist) the *gradient* and the *Hessian* of f, respectively, and use the following notation for them:

$$\nabla f \coloneqq Df, \qquad \nabla^2 f \coloneqq D^2 f.$$

In this particular case, according to (2.1.7) and the definition of $\mathbb{E}^*$, for any $x \in \operatorname{int} Q$, for which the corresponding derivatives exist, we have

$$\nabla f(x) \in \mathbb{E}^*, \qquad \nabla^2 f(x) \in \mathcal{L}(\mathbb{E}, \mathbb{E}^*).$$

Also, by (2.1.8) and the definition of $\langle \cdot, \cdot \rangle$, for all $h, h_1, h_2 \in \mathbb{E}$, we have

$$Df(x)[h] = \langle \nabla f(x), h \rangle, \qquad D^2 f(x)[h_1, h_2] = \langle \nabla^2 f(x) h_1, h_2 \rangle. \tag{2.1.9}$$

Since the form $D^2 f(x)[\cdot, \cdot]$ is symmetric, the Hessian is actually self-adjoint:

$$\nabla^2 f(x) \in \mathcal{S}(\mathbb{E}, \mathbb{E}^*).$$

In the case $\mathbb{E} = \mathbb{E}^* = \mathbb{R}^n$, the gradient and Hessian are given by the vector of partial derivatives and the (symmetric) matrix of second-order partial derivatives, respectively: $\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_i}\right)_{i=1}^n$ and $\nabla^2 f(x) = \left(\frac{\partial^2 f(x)}{\partial x_i \partial x_j}\right)_{i,j=1}^n$.

2.1.5 Norms

Any norm $\|\cdot\|$ in $\mathbb{E}$ naturally induces the following[4] *conjugate norm* in $\mathbb{E}^*$:

$$\|s\|_* \coloneqq \max_h \{ |\langle s, h \rangle| : \|h\| = 1 \}, \qquad \forall s \in \mathbb{E}^*. \tag{2.1.10}$$

[4] The absolute value in (2.1.10) is optional, as we can always replace h with $-h$.

This definition ensures the *Cauchy–Schwarz inequality*:

$$|\langle s, h\rangle| \le \|s\|_* \|h\|, \qquad \forall s \in \mathbb{E}^*,\ \forall h \in \mathbb{E}. \tag{2.1.11}$$

In this thesis, we usually work with *Euclidean norms*. Any such a norm is generated by a certain operator $B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ according to

$$\|h\|_B \coloneqq \langle Bh, h\rangle^{1/2}, \qquad \forall h \in \mathbb{E}. \tag{2.1.12}$$

The conjugate norm for $\|\cdot\|_B$ is also Euclidean and is generated by B^{-1}:

$$\|s\|_B^* = \langle s, B^{-1}s\rangle^{1/2}, \qquad \forall s \in \mathbb{E}^*. \tag{2.1.13}$$

Furthermore, the Cauchy–Schwartz inequality (2.1.11) becomes an equality iff s and Bh are collinear.

In particular, when $\mathbb{E} = \mathbb{E}^* = \mathbb{R}^n$ and $B = I$ (the identity matrix), both (2.1.12) and (2.1.13) coincide with the standard Euclidean norm in $\mathbb{R}^n$.

Given a function $f\colon Q \to \mathbb{R}$ defined and twice differentiable on an open set $Q \subseteq \mathbb{E}$, and a point $x \in Q$ with $\nabla^2 f(x) \succ 0$, we prefer to use the following notation:

$$\|\cdot\|_x \coloneqq \|\cdot\|_{\nabla^2 f(x)}, \qquad \|\cdot\|_x^* \coloneqq \|\cdot\|_{\nabla^2 f(x)}^*, \tag{2.1.14}$$

provided that there is no ambiguity with the reference function f.

Recall from (2.1.7) that, for a function $f\colon Q \to \mathbb{R}$ defined on a set $Q \subseteq \mathbb{E}$ and differentiable as many times as necessary at a point $x \in \operatorname{int} Q$, the derivatives of f at x are certain linear operators:

$$D^p f(x) \in \mathcal{L}_p, \qquad p \ge 1,$$

where $\mathcal{L}_p$ is a certain vector space, defined recursively by

$$\mathcal{L}_0 \coloneqq \mathbb{R}, \qquad \mathcal{L}_p \coloneqq \mathcal{L}(\mathbb{E}, \mathcal{L}_{p-1}), \qquad p \ge 1. \tag{2.1.15}$$

Therefore, to measure the size of derivatives of f and of their various combinations, we need to define a suitable norm on the space $\mathcal{L}_p$ for any $p \ge 1$.

Any norm $\|\cdot\|$ in $\mathbb{E}$ naturally induces a certain *operator norm* in $\mathcal{L}_p$ for any $p \ge 1$. We have already seen the corresponding definition for the space $\mathcal{L}_1 \equiv \mathbb{E}^*$—this is the conjugate norm (2.1.10). Similarly, we can

recursively define[5], for any $p \geq 1$ and any $M_p \in \mathcal{L}_p$,

$$\|M_p\| \coloneqq \max_{h \in \mathbb{E}} \{\|M_p h\| : \|h\| = 1\}, \tag{2.1.16}$$

with the convention that $\|t\| \coloneqq |t|$ for any $t \in \mathbb{R}$. This definition is compatible with that in (2.1.10) (when $p = 1$), and ensures that, for any integer $p \geq 1$, any $M_p \in \mathcal{L}_p$ and any $h \in \mathbb{E}$, we have the following inequality:

$$\|M_p h\| \leq \|M_p\| \|h\|. \tag{2.1.17}$$

Unrolling recursive definition (2.1.16), we come to the following more explicit expression that works for any $p \geq 1$ and any $M_p \in \mathcal{L}_p$:

$$\|M_p\| = \max_{h_1, \ldots, h_p \in \mathbb{E}} \{|M_p[h_1, \ldots, h_p]| : \|h_1\| = \cdots = \|h_p\| = 1\}, \tag{2.1.18}$$

where $M_p[h_1, \ldots, h_p] \coloneqq M_p h_1 \ldots h_p \in \mathbb{R}$ is the multilinear form induced by the operator M_p. In the important special case when the form $M_p[\cdot, \ldots, \cdot]$ is *symmetric* and the norm $\|\cdot\|$ in $\mathbb{E}$ is *Euclidean*, it turns out that the maximum in (2.1.18) can be achieved when all $h_1, \ldots, h_p$ coincide, and therefore the following simpler expression could be used (see Appendix 1 in [139]):

$$\|M_p\| = \max_{h \in \mathbb{E}} \{|M_p[h]^p| : \|h\| = 1\}. \tag{2.1.19}$$

Throughout this thesis, we always assume that the norm in the operator spaces (2.1.15) is exactly the operator norm defined in (2.1.18), until explicitly stated otherwise.

In the particular case when $p = 2$ and $\mathbb{E} = \mathbb{E}^* = \mathbb{R}^n$ (with the standard choice of $\|\cdot\|$), the norm, defined in (2.1.16), is the standard spectral norm of a matrix (maximal singular value).

2.1.6 Relative Eigenvalues and Eigenvectors

Given two linear operators $A \in \mathcal{S}(\mathbb{E}, \mathbb{E}^*)$ and $B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$, and a scalar $\lambda \in \mathbb{R}$, we call λ a *(relative) eigenvalue* of A w.r.t. B (also known as

[5] Note that we use the same notation $\|\cdot\|$ to denote different norms in different spaces. However, in most cases, this should not cause any problems since the precise meaning of the norm can usually be inferred from the context based on the "type" of the variable. For example, since $M_p \in \mathcal{L}_p$, $\|M_p\|$ is the norm in the space $\mathcal{L}_p$; since $M_p h \in \mathcal{L}_{p-1}$, $\|M_p h\|$ is the norm in the space $\mathcal{L}_{p-1}$; since $h \in \mathbb{E}$, $\|h\|$ is the norm in the space $\mathbb{E}$, etc. Should there arise any ambiguity, we can always clarify the meaning of the norm either with words, or by adding the particular space as an index of the norm (e.g., $\|\cdot\|_{\mathcal{L}_p}$).

a *generalized eigenvalue* or an *eigenvalue of the pencil* (A, B) in some texts) if there exists $x \in \mathbb{E} \setminus \{0\}$ such that

$$Ax = \lambda Bx. \tag{2.1.20}$$

The vector x in this definition is called a *(relative) eigenvector* of A w.r.t. B corresponding to the (relative) eigenvalue λ. The set of all eigenvalues of A w.r.t. B is referred to as a *(relative) spectrum* of A w.r.t. B.

Of course, each relative eigenvalue λ of A w.r.t. B is a standard eigenvalue of $B^{-1}A \in \mathcal{L}(\mathbb{E}, \mathbb{E})$, and vice versa. However, usually, it is better to keep A and B separately, as the operator $B^{-1}A$ is, in general, not self-adjoint.

In the special case $\mathbb{E} = \mathbb{E}^* = \mathbb{R}^n$ (with the standard identification of linear operators with matrices), the eigenvalues of A w.r.t. B are precisely the usual eigenvalues of the matrix $B^{-1/2}AB^{-1/2} \in \mathbb{S}^n$, where $B^{-1/2} \in \mathbb{S}^n_{++}$ is the inverse square root of B.

From (2.1.5) and (2.1.20), it is easy to see that there is a direct correspondence between operator inequalities and uniform bounds on the relative spectrum: for any $A \in \mathcal{S}(\mathbb{E}, \mathbb{E}^*)$, $B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ and $\alpha \in \mathbb{R}$, we have

$$\alpha B \preceq A \quad [\text{resp. } A \preceq \alpha B] \qquad \Longleftrightarrow \qquad \alpha \leq \lambda \quad [\text{resp. } \lambda \leq \alpha]$$

for all eigenvalues λ of A w.r.t. B. Similar relationships hold for strict inequalities.

The following result from Linear Algebra is of fundamental importance. It shows that, for any linear operators $A \in \mathcal{S}(\mathbb{E}, \mathbb{E}^*)$ and $B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$, it is possible to construct an orthonormal basis in the space $\mathbb{E}$ consisting entirely of eigenvectors of A w.r.t. B.

Proposition 2.1.2 (Spectral theorem)**.** *Let $A \in \mathcal{S}(\mathbb{E}, \mathbb{E}^*)$, $B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$, and let $n \coloneqq \dim \mathbb{E}$. Then, there exist $\lambda_1, \dots, \lambda_n \in \mathbb{R}$ and $u_1, \dots, u_n \in \mathbb{E}$, such that*

$$Au_i = \lambda_i Bu_i, \qquad 1 \leq i \leq n,$$

and

$$\langle Bu_i, u_j \rangle = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j, \end{cases} \qquad 1 \leq i, j \leq n.$$

Proof. Although this particular formulation of the spectral theorem is not common, it immediately follows from a more classical one, e.g., the spectral theorem for an operator in a Euclidean space (see Theorem 2.8.5 in [99]).

Indeed, by introducing the inner product $(x, y) \coloneqq \langle Bx, y\rangle$, we can convert $\mathbb{E}$ into a Euclidean space, in which the operator $S \coloneqq B^{-1}A \in \mathcal{L}(\mathbb{E}, \mathbb{E})$ is actually self-adjoint (i.e., $(Sx, y) = (x, Sy)$ for all $x, y \in \mathbb{E}$), and then apply the standard spectral theorem. □

The eigenvalues $\lambda_1, \ldots, \lambda_n$, given by Proposition 2.1.2, are, in fact, unique (up to a reordering), and are referred to as the *(relative) eigenvalues* of A w.r.t. B. Note, however, that $\lambda_1, \ldots, \lambda_n$ are not necessarily distinct, as certain eigenvalues may be repeated multiple times according to their *multiplicity*.

2.1.7 Trace Product

For any linear operator $S\colon \mathbb{E} \to \mathbb{E}$, the *trace* of S, denoted by $\operatorname{tr}(S)$ $(\in \mathbb{R})$, is defined as the trace of the matrix representation of S w.r.t. an arbitrarily chosen basis in $\mathbb{E}$ (recall that the trace of a square matrix is the sum of its diagonal elements). It is important that the result is independent of the particular choice of the basis since different bases generate similar matrices (for more details, see Section 1.4.9 in [99]).

Note that the same construction does not work properly for a more general linear operator which acts from one vector space to another. For example, if one attempts to define the trace of a linear operator $A\colon \mathbb{E} \to \mathbb{E}^*$ as the trace of the matrix representation of A w.r.t. an arbitrarily chosen pair of bases in $\mathbb{E}$ and $\mathbb{E}^*$, then the result will no longer be basis-independent (even if the basis in $\mathbb{E}^*$ is dual[6] to the one in $\mathbb{E}$).

Nevertheless, given two linear operators $H\colon \mathbb{E}_1^* \to \mathbb{E}_2$ and $A\colon \mathbb{E}_1 \to \mathbb{E}_2^*$, we can properly define their *trace product*:

$$\langle H, A\rangle \coloneqq \operatorname{tr}(H^*A). \tag{2.1.21}$$

The operator H^*A in this definition acts from $\mathbb{E}_1$ to $\mathbb{E}_1$, therefore, its trace is well-defined and is basis-independent.

When $\mathbb{E}_1 = \mathbb{E}_1^* = \mathbb{R}^n$ and $\mathbb{E}_2 = \mathbb{E}_2^* = \mathbb{R}^m$ (with the standard identification of linear operators with matrices), the trace product is the usual Frobenius inner product: $\langle H, A\rangle = \operatorname{tr}(H^TA)$.

[6] Recall that a basis $f \coloneqq (f_1, \ldots, f_n)$ in $\mathbb{E}^*$ is called *dual* to a basis $e \coloneqq (e_1, \ldots, e_n)$ in $\mathbb{E}$ if, for any $1 \le i, j \le n$, we have $\langle f_i, e_j\rangle = \delta_{i,j}$, where $\delta_{i,j} = 1$ whenever $i = j$, and $\delta_{i,j} = 0$ whenever $i \ne j$. In fact, for any basis e in $\mathbb{E}$, the dual basis is unique and given by $\langle f_i, x\rangle \coloneqq \bar{x}_i$ for any $x \in \mathbb{E}$ and $1 \le i \le n$, where $\bar{x}_i$ is the ith coordinate of the vector x in the basis e (see Section 1.3.9 in [99]).

Observe that the mapping $\langle \cdot, \cdot \rangle$, defined in (2.1.21), is a nondegenerate bilinear form on $\mathcal{L}(\mathbb{E}_1^*, \mathbb{E}_2) \times \mathcal{L}(\mathbb{E}_1, \mathbb{E}_2^*)$. Therefore, it allows us to make the following identification of the dual space of the linear operator space:

$$[\mathcal{L}(\mathbb{E}_1, \mathbb{E}_2^*)]^* = \mathcal{L}(\mathbb{E}_1^*, \mathbb{E}_2), \tag{2.1.22}$$

in the sense that the transformation that sends each $H \in \mathcal{L}(\mathbb{E}_1^*, \mathbb{E}_2)$ into $\tilde{H} \in [\mathcal{L}(\mathbb{E}_1, \mathbb{E}_2^*)]^*$, defined by $\langle \tilde{H}, A \rangle = \langle H, A \rangle$ for all $A \in \mathcal{L}(\mathbb{E}_1, \mathbb{E}_2^*)$, is a bijection.

Inheriting the bilinear form $\langle \cdot, \cdot \rangle$, defined in (2.1.21), onto the subspace $\mathcal{S}(\mathbb{E}^*, \mathbb{E}) \times \mathcal{S}(\mathbb{E}, \mathbb{E}^*)$, we can also make the following identification:

$$[\mathcal{S}(\mathbb{E}, \mathbb{E}^*)]^* = \mathcal{S}(\mathbb{E}^*, \mathbb{E}), \tag{2.1.23}$$

which is consistent with that from (2.1.22).

Proposition 2.1.3. *The trace product has the following properties:*

(i) *For any $A \in \mathcal{L}(\mathbb{E}_1, \mathbb{E}_2^*)$ and any $x_1 \in \mathbb{E}_1$, $x_2 \in \mathbb{E}_2$,*

$$\langle Ax_1, x_2 \rangle = \langle x_2 x_1^*, A \rangle. \tag{2.1.24}$$

(ii) *If $A \in \mathcal{S}(\mathbb{E}, \mathbb{E}^*)$ is invertible, then*

$$\langle A^{-1}, A \rangle = n, \tag{2.1.25}$$

where $n = \dim \mathbb{E}$.

(iii) *For any $A \in \mathcal{S}_+(\mathbb{E}, \mathbb{E}^*)$ and any $H_1, H_2 \in \mathcal{S}(\mathbb{E}^*, \mathbb{E})$,*

$$H_1 \preceq H_2 \implies \langle H_1, A \rangle \leq \langle H_2, A \rangle.$$

Similarly, for any $H \in \mathcal{S}_+(\mathbb{E}^*, \mathbb{E})$ and any $A_1, A_2 \in \mathcal{S}(\mathbb{E}, \mathbb{E}^*)$,

$$A_1 \preceq A_2 \implies \langle H, A_1 \rangle \leq \langle H, A_2 \rangle.$$

(iv) *If $A \in \mathcal{S}(\mathbb{E}, \mathbb{E}^*)$ and $B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$, then*

$$\langle B^{-1}, A \rangle = \sum_{i=1}^{n} \lambda_i,$$

where $n = \dim \mathbb{E}$ and $\lambda_1, \ldots, \lambda_n \in \mathbb{R}$ are the eigenvalues of A w.r.t. B.

Proof. All these properties follow directly from their matrix counterparts.

For example, let us show in detail how to prove (i). Denote

$$H \coloneqq x_2 x_1^* \in \mathcal{L}(\mathbb{E}_1^*, \mathbb{E}_2). \tag{2.1.26}$$

Then, according to (2.1.21), we have

$$\langle x_2 x_1^*, A\rangle = \langle H, A\rangle = \operatorname{tr}(H^* A). \tag{2.1.27}$$

Denote $n_1 \coloneqq \dim \mathbb{E}_1$ and $n_2 \coloneqq \dim \mathbb{E}_2$. Let us fix arbitrary bases e_1 and e_2 in the spaces $\mathbb{E}_1$ and $\mathbb{E}_2$, respectively, and let f_1 and f_2 be the corresponding dual bases in $\mathbb{E}_1^*$ and $\mathbb{E}_2^*$, respectively. Let $\bar{H}, \bar{A} \in \mathbb{R}^{n_2 \times n_1}$ be the matrices of the operators H and A, respectively, in the pairs of bases (f_1, e_2) and (e_1, f_2), respectively. From Linear Algebra, it is known that the matrix of the adjoint operator in a pair of dual bases is the transpose of the corresponding matrix of the operator itself (see Section 1.7.4 in [99]). Therefore, the matrix of $H^* \in \mathcal{L}(\mathbb{E}_2^*, \mathbb{E}_1)$ in the pair of bases (f_2, e_1) is $\bar{H}^T \in \mathbb{R}^{n_1 \times n_2}$. Also, it is known that the matrix of the composition of linear operators is the product of the corresponding matrices (see, Section 1.4.7 in [99]). Hence, the matrix of $H^* A \in \mathcal{L}(\mathbb{E}_1, \mathbb{E}_1)$ is $\bar{H}^T \bar{A} \in \mathbb{R}^{n_1 \times n_1}$. Thus, according to our definition of trace, we have

$$\operatorname{tr}(H^* A) = \operatorname{tr}(\bar{H}^T \bar{A}), \tag{2.1.28}$$

where the trace in the right-hand side is the standard matrix trace.

Let $\bar{x}_1 \in \mathbb{R}^{n_1}$ and $\bar{x}_2 \in \mathbb{R}^{n_2}$ be the coordinate representation of x_1 and x_2 in the bases e_1 and e_2, respectively. It is not difficult to see from (2.1.26) that the matrix of the operator H is given by

$$\bar{H} = \bar{x}_2 \bar{x}_1^T.$$

Hence, by standard matrix calculus,

$$\begin{aligned} \operatorname{tr}(\bar{H}^T \bar{A}) &= \operatorname{tr}([\bar{x}_2 \bar{x}_1^T]^T \bar{A}) = \operatorname{tr}(\bar{x}_1 \bar{x}_2^T \bar{A}) \\ &= \operatorname{tr}(\bar{x}_2^T \bar{A} \bar{x}_1) = \langle \bar{A}\bar{x}_1, \bar{x}_2\rangle_{\mathbb{R}^n} = \langle \bar{s}_2, \bar{x}_2\rangle_{\mathbb{R}^n}, \end{aligned} \tag{2.1.29}$$

where

$$\bar{s}_2 \coloneqq \bar{A}\bar{x}_1 \in \mathbb{R}^{n_2},$$

and $\langle \cdot, \cdot \rangle_{\mathbb{R}^n}$ is the standard dot product in $\mathbb{R}^n$ (see (2.1.1)).

From Linear Algebra, we know that $\bar{s}_2$ is exactly the coordinate repre-

sentation of

$$s_2 \coloneqq Ax_1 \in \mathbb{E}_2^*$$

in the dual basis f_2 (see Section 1.4.4 in [99]). Also, we know that the dual pairing $\langle \cdot, \cdot \rangle$ corresponds to the standard dot product of the coordinate representations w.r.t. an arbitrary choice of the pair of dual bases (see Section 1.7.3 in [99]). Therefore,

$$\langle \bar{s}_2, \bar{x}_2 \rangle_{\mathbb{R}^n} = \langle s_2, x_2 \rangle = \langle Ax_1, x_2 \rangle. \tag{2.1.30}$$

Putting together (2.1.27)–(2.1.30), we obtain (2.1.24). The other properties can be proved in the same way. □

2.1.8 Determinant Product

Another important characteristic of a linear operator $S\colon \mathbb{E} \to \mathbb{E}$ is its *determinant*, denoted by $\det(S)$ ($\in \mathbb{R}$), and defined as the determinant of the matrix representation of S w.r.t. an arbitrarily chosen basis in $\mathbb{E}$. Similarly to the trace, the determinant of such an operator is basis-independent (see Section 1.4.9 in [99]).

For any linear operators $H\colon \mathbb{E}^* \to \mathbb{E}$ and $A\colon \mathbb{E} \to \mathbb{E}^*$, we can define their *determinant product*[7] by

$$\det(H, A) \coloneqq \det(HA). \tag{2.1.31}$$

The operator HA in this definition acts from $\mathbb{E}$ to $\mathbb{E}$, therefore its determinant is well-defined and is basis-independent.

In the special case when $\mathbb{E} = \mathbb{E}^* = \mathbb{R}^n$ (with the standard identification of operators with matrices), the determinant product actually corresponds to the product of determinants: $\det(H, A) = \det(H)\det(A)$. In general, however, this formula makes no sense, as there are no well-defined notions of $\det(H)$ and $\det(A)$ for $H \in \mathcal{L}(\mathbb{E}^*, \mathbb{E})$ and $A \in \mathcal{L}(\mathbb{E}, \mathbb{E}^*)$.

Proposition 2.1.4. *The determinant product has the following properties:*

(i) *If $A \in \mathcal{L}(\mathbb{E}, \mathbb{E}^*)$ is invertible, then*

$$\det(A^{-1}, A) = 1.$$

[7] In principle, this definition can be introduced for a more general pair of linear operators $H\colon \mathbb{E}_2^* \to \mathbb{E}_1$ and $A\colon \mathbb{E}_1 \to \mathbb{E}_2^*$. However, in this thesis, we will be interested only in the particular case when $\mathbb{E}_1 = \mathbb{E}_2$.

(ii) *For any* $H \in \mathcal{L}(\mathbb{E}^*, \mathbb{E})$*, any* $A \in \mathcal{L}(\mathbb{E}, \mathbb{E}^*)$*, and any* $\delta \in \mathbb{R}$*,*

$$\det(H, \delta A) = \det(\delta H, A) = \delta^n \det(H, A),$$

where $n = \dim \mathbb{E}$*.*

(iii) *For any* $H_1, H_2 \in \mathcal{L}(\mathbb{E}^*, \mathbb{E})$ *and any* $A_1, A_2 \in \mathcal{L}(\mathbb{E}, \mathbb{E}^*)$*,*

$$\det(H_1, A_1) \det(H_2, A_2) = \det(H_1, A_1 H_2 A_2) = \det(H_1 A_1 H_2, A_2).$$

(iv) *If* $H \in \mathcal{L}(\mathbb{E}^*, \mathbb{E})$ *and* $A \in \mathcal{L}(\mathbb{E}, \mathbb{E}^*)$ *are invertible, then*

$$\det(A^{-1}, H^{-1}) = [\det(H, A)]^{-1}$$

with $\det(H, A) \neq 0$*.*

(v) *For any* $A \in \mathcal{S}_+(\mathbb{E}, \mathbb{E}^*)$ *and any* $H_1, H_2 \in \mathcal{S}(\mathbb{E}^*, \mathbb{E})$*,*

$$H_1 \preceq H_2 \implies \det(H_1, A) \leq \det(H_2, A).$$

Similarly, for any $H \in \mathcal{S}_+(\mathbb{E}^*, \mathbb{E})$ *and any* $A_1, A_2 \in \mathcal{S}(\mathbb{E}, \mathbb{E}^*)$*,*

$$A_1 \preceq A_2 \implies \det(H, A_1) \leq \det(H, A_2).$$

(vi) *If* $A \in \mathcal{S}(\mathbb{E}, \mathbb{E}^*)$ *and* $B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$*, then*

$$\det(B^{-1}, A) = \prod_{i=1}^{n} \lambda_i,$$

where $n = \dim \mathbb{E}$ *and* $\lambda_1, \ldots, \lambda_n \in \mathbb{R}$ *are the eigenvalues of* A *w.r.t.* B*.*

Proof. All the properties are simple extensions of their matrix counterparts, and can be justified in the same way as in the proof of Proposition 2.1.3. □

For computing determinant products involving rank-one perturbations, the following result is often useful.

Proposition 2.1.5. *Let* $A \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$*,* $s \in \mathbb{E}^*$ *and* $\alpha \in \mathbb{R}$*. Then,*

$$\det(A^{-1}, A + \alpha s s^*) = 1 + \alpha \langle s, A^{-1} s \rangle.$$

Proof. Indeed, the operator $A+\alpha ss^*$ has the following eigenvalues w.r.t. A: $\lambda_1 = 1+\alpha\langle s, A^{-1}s\rangle$ (with the eigenvector $A^{-1}s$) and $\lambda_2 = \cdots = \lambda_n = 1$. It remains to apply Proposition 2.1.4(vi). □

2.1.9 Relative Volume

Given two compact sets $Q, Q_0 \subseteq \mathbb{E}$ with $\operatorname{int} Q_0 \neq \varnothing$, we can define the *relative volume* of Q w.r.t. Q_0 as the ratio of the volumes of their coordinate representations w.r.t. to an arbitrarily selected basis:

$$\operatorname{vol}(Q/Q_0) \coloneqq \frac{\operatorname{vol}\bar{Q}}{\operatorname{vol}\bar{Q}_0}, \tag{2.1.32}$$

where $\bar{Q}, \bar{Q}_0 \subseteq \mathbb{R}^n$ are the coordinate representations of Q and Q_0, respectively, in a certain (arbitrarily chosen) basis in $\mathbb{E}$, and $\operatorname{vol}\bar{Q}$ and $\operatorname{vol}\bar{Q}_0$ are the standard volumes / Lebesgue measures (in $\mathbb{R}^n$) of $\bar{Q}$ and $\bar{Q}_0$, respectively. Note that $\operatorname{vol}\bar{Q}_0 \neq 0$ since Q_0 is assumed to have a nonempty interior.

Contrary to the "usual volume $\operatorname{vol} Q$", which could be tentatively defined as the volume of the coordinate representation of Q w.r.t. an arbitrarily chosen basis, the relative volume is a basis-independent notion.

Proposition 2.1.6. *The relative volume, defined in* (2.1.32)*, is independent of the particular choice of basis.*

Proof. Indeed, let e and e' be two bases in $\mathbb{E}$. Let $\bar{Q}^e, \bar{Q}_0^e, \bar{Q}^{e'}, \bar{Q}_0^{e'} \subseteq \mathbb{R}^n$ be the coordinate representations of Q and Q_0 in the bases e and e', respectively. From Linear Algebra, we know that these coordinate representations are linked as follows[8]:

$$\bar{Q}^e = T^e_{e'}\bar{Q}^{e'}, \qquad \bar{Q}_0^e = T^e_{e'}\bar{Q}_0^{e'},$$

where $T^e_{e'} \in \mathbb{R}^{n\times n}$ is the corresponding change-of-basis (nondegenerate) matrix (see Section 1.4.8 in [99]). Hence, by the standard formula for the volume change under a linear transformation (in $\mathbb{R}^n$), we have

$$\operatorname{vol}\bar{Q}^e = |\det T^e_{e'}|(\operatorname{vol}\bar{Q}^{e'}), \qquad \operatorname{vol}\bar{Q}_0^e = |\det T^e_{e'}|(\operatorname{vol}\bar{Q}_0^{e'}),$$

[8]Here we use the following standard notation: for a matrix $T \in \mathbb{R}^{n\times n}$ and a set $\bar{Q} \subseteq \mathbb{R}^n$, by $T\bar{Q} \coloneqq \{Tx : x \in \bar{Q}\} \subseteq \mathbb{R}^n$, we denote the image of $\bar{Q}$ under the linear transformation defined by T.

where $\det T_{e'}^{e}$ is the usual matrix determinant. Thus, we conclude that $\operatorname{vol}\bar{Q}^{e} / \operatorname{vol}\bar{Q}_0^{e} = \operatorname{vol}\bar{Q}^{e'} / \operatorname{vol}\bar{Q}_0^{e'}$. □

Let us state some basic properties of the relative volume.

Proposition 2.1.7. *Let $Q, Q_0 \subseteq \mathbb{E}$ be compact sets with $\operatorname{int} Q_0 \neq \varnothing$. Then:*

(i) $\operatorname{vol}(Q/Q_0)$ *is invariant w.r.t. arbitrary translations of Q and Q_0.*

(ii) *For any compact set $Q' \subseteq \mathbb{E}$:*

$$Q \subseteq Q' \implies \operatorname{vol}(Q/Q_0) \leq \operatorname{vol}(Q'/Q_0).$$

(iii) *If $\operatorname{int} Q \neq \varnothing$, then*

$$\operatorname{vol}(Q/Q) = 1.$$

(iv) *For any $\delta > 0$, we have*

$$\operatorname{vol}\big((\delta Q)/Q_0\big) = \operatorname{vol}\big(Q/(\delta^{-1}Q_0)\big) = \delta^n \operatorname{vol}(Q/Q_0),$$

where $n = \dim \mathbb{E}$.

(v) *For any compact set $Q_1 \subseteq \mathbb{E}$ with $\operatorname{int} Q_1 \neq \varnothing$, we have*

$$\operatorname{vol}(Q/Q_0) = \operatorname{vol}(Q/Q_1) \operatorname{vol}(Q_1/Q_0). \tag{2.1.33}$$

Proof. All the properties are rather straightforward consequences of definition (2.1.32). For instance, let us show how to justify Proposition 2.1.7(v). For this, let us fix an arbitrary basis e in $\mathbb{E}$. Let $\bar{Q}, \bar{Q}_0, \bar{Q}_1 \subseteq \mathbb{R}^n$ be the coordinate representations of Q, Q_0 and Q_1, respectively, in the basis e. Then, according to (2.1.32),

$$\operatorname{vol}(Q/Q_0) = \frac{\operatorname{vol}\bar{Q}}{\operatorname{vol}\bar{Q}_0} = \frac{\operatorname{vol}\bar{Q}}{\operatorname{vol}\bar{Q}_1} \frac{\operatorname{vol}\bar{Q}_1}{\operatorname{vol}\bar{Q}_0} = \operatorname{vol}(Q/Q_1) \operatorname{vol}(Q_1/Q_0),$$

which is exactly (2.1.33). The other properties can be proved similarly. □

The following result will be particularly important for us.

Proposition 2.1.8. *Let Q and Q_0 be two ellipsoids:*

$$Q \coloneqq \{x \in \mathbb{E} : \|x - \hat{x}\|_G \leq 1\}, \qquad Q_0 \coloneqq \{x \in \mathbb{E} : \|x - \hat{x}_0\|_{G_0} \leq 1\},$$

where $\hat{x}, \hat{x}_0 \in \mathbb{E}$ *and* $G, G_0 \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$. *Then,*

$$\operatorname{vol}(Q/Q_0) = [\det(G_0^{-1}, G)]^{-1/2} = [\det(G^{-1}, G_0)]^{1/2}. \tag{2.1.34}$$

Proof. In view of Proposition 2.1.7(i), we can assume that $\hat{x} = \hat{x}_0 = 0$. Thus, according to (2.1.12),

$$Q = \{x \in \mathbb{E} : \langle Gx, x\rangle \le 1\}, \qquad Q_0 = \{x \in \mathbb{E} : \langle G_0 x, x\rangle \le 1\}. \tag{2.1.35}$$

Let us fix (arbitrarily) a pair of dual bases (e, f) in the spaces $\mathbb{E}$ and $\mathbb{E}^*$. Let $\bar{G}, \bar{G}_0 \in \mathbb{R}^{n\times n}$ be the corresponding matrix representations of G and G_0, respectively ($n \coloneqq \dim \mathbb{E}$). Note that the matrices $\bar{G}$ and $\bar{G}_0$ are symmetric and positive definite: $\bar{G}, \bar{G}_0 \in \mathbb{S}^n_{++}$ (since $G, G_0 \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$). Further, for any $x \in \mathbb{E}$, we have $\langle Gx, x\rangle = \langle \bar{G}\bar{x}, \bar{x}\rangle_{\mathbb{R}^n}$ and $\langle G_0 x, x\rangle = \langle \bar{G}_0\bar{x}, \bar{x}\rangle_{\mathbb{R}^n}$, where $\bar{x} \in \mathbb{R}^n$ is the coordinate representation of x in the basis e, and $\langle\cdot,\cdot\rangle_{\mathbb{R}^n}$ is the standard dot product in $\mathbb{R}^n$ (defined in (2.1.1)). Using this observation, it is easy to see from (2.1.35) that the ellipsoids Q and Q_0 have, respectively, the following matrix representations in the basis e:

$$\bar{Q} = \{\bar{x} \in \mathbb{R}^n : \langle \bar{G}\bar{x}, \bar{x}\rangle_{\mathbb{R}^n} \le 1\}, \qquad \bar{Q}_0 = \{\bar{x} \in \mathbb{R}^n : \langle \bar{G}_0\bar{x}, \bar{x}\rangle_{\mathbb{R}^n} \le 1\}.$$

Let $\bar{G}^{1/2}, \bar{G}_0^{1/2} \in \mathbb{S}^n_{++}$ be the matrix square roots of $\bar{G}$ and $\bar{G}_0$, respectively, and let $\bar{B}$ be the standard Euclidean ball in $\mathbb{R}^n$:

$$\bar{B} \coloneqq \{\bar{x} \in \mathbb{R}^n : \|\bar{x}\|_{\mathbb{R}^n} \le 1\},$$

where $\|\cdot\|_{\mathbb{R}^n} \coloneqq \langle\cdot,\cdot\rangle^{1/2}_{\mathbb{R}^n}$ is the standard Euclidean norm in $\mathbb{R}^n$. Using this notation, we can represent each of the sets $\bar{Q}$ and $\bar{Q}_0$ as the image of $\bar{B}$ under a linear transformation:

$$\bar{Q} = \bar{G}^{-1/2}\bar{B}, \qquad \bar{Q}_0 = \bar{G}_0^{-1/2}\bar{B},$$

where $\bar{G}^{-1/2}$ and $\bar{G}_0^{-1/2}$ are the inverse matrices of $\bar{G}^{1/2}$ and $\bar{G}_0^{1/2}$.

Applying now the classical formula for the volume change under a linear transformation (in $\mathbb{R}^n$), we obtain

$$\operatorname{vol}\bar{Q} = \det(\bar{G}^{-1/2})\operatorname{vol}\bar{B}, \qquad \operatorname{vol}\bar{Q}_0 = \det(\bar{G}_0^{-1/2})\operatorname{vol}\bar{B}.$$

Combining this with (2.1.32) and using standard matrix calculus, we get

$$\begin{aligned}\operatorname{vol}(Q/Q_0) = \frac{\operatorname{vol}\bar{Q}}{\operatorname{vol}\bar{Q}_0} = \frac{\det(\bar{G}^{-1/2})}{\det(\bar{G}_0^{-1/2})} = \left[\frac{\det\bar{G}}{\det\bar{G}_0}\right]^{-1/2} \\ = [\det(\bar{G}_0^{-1}\bar{G})]^{-1/2} = [\det(G_0^{-1}, G)]^{-1/2},\end{aligned}$$

where the last identity is due to definition (2.1.31) and the fact that $\bar{G}_0^{-1}$ is exactly the matrix of the operator G_0^{-1} w.r.t. our chosen pair of bases (f, e) (see Section 1.4.7 in [99]). This proves the first identity in (2.1.34). The second identity follows from the first one using Proposition 2.1.4(iv). □

2.2 Standard Function Classes

Let us review some standard function classes which we will be using throughout this thesis.

2.2.1 Convex Functions

Most of the time, we be working with convex functions and convex sets.

Definition 2.2.1 (Convex set)**.** A set $Q \subseteq \mathbb{E}$ is called *convex* if, for all $x, y \in Q$ and all $\alpha \in [0, 1]$, we have

$$(1 - \alpha)x + \alpha y \in Q.$$

Definition 2.2.2 (Convex function)**.** A function $f\colon Q \to \mathbb{R}$, defined on a convex set $Q \subseteq \mathbb{E}$, is called *convex* (on Q) if, for any $x, y \in Q$ and any $\alpha \in [0, 1]$, we have

$$f\big((1 - \alpha)x + \alpha y\big) \leq (1 - \alpha)f(x) + \alpha f(y).$$

If this inequality is strict whenever $x \neq y$ and $\alpha \in (0, 1)$, then the function f is called *strictly convex*.

For differentiable functions, we have the following equivalent characterizations of convexity (see Definition 2.1.2 and Theorems 2.1.2–2.1.4 in [133]).

Proposition 2.2.3. *Let $f\colon Q \to \mathbb{R}$ be a function, where $Q \subseteq \mathbb{E}$ is an open convex set.*

(i) *Suppose f is differentiable on Q. Then, f is convex on Q iff, for any $x, y \in Q$, we have*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle.$$

(ii) *Suppose f is differentiable on Q. Then, f is convex on Q iff, for any $x, y \in Q$, we have*

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq 0.$$

(iii) *Suppose f is twice differentiable on Q. Then, f is convex on Q iff, for any $x \in Q$, we have*

$$\nabla^2 f(x) \succeq 0.$$

Sometimes, it is convenient to work with *extended real-valued convex functions*, which are defined on the whole space but are allowed to take infinite values at certain points.

A function $f \colon \mathbb{E} \to \mathbb{R} \cup \{+\infty\}$ is called *convex* if its *effective domain*

$$\operatorname{dom} f \coloneqq \{x \in \mathbb{E} : f(x) < +\infty\}$$

is a convex set, and the restriction of f onto $\operatorname{dom} f$ is a convex function in the sense of Definition 2.2.2: for all $x, y \in \operatorname{dom} f$ and all $\alpha \in [0, 1]$, we have

$$f\big((1-\alpha)x + \alpha y\big) \leq (1-\alpha) f(x) + \alpha f(y).$$

Clearly, every real-valued convex function $f \colon Q \to \mathbb{R}$, defined on a convex set $Q \subseteq \mathbb{E}$, can always be treated (by a slight abuse of notation) as an extended real-valued convex function $f \colon \mathbb{E} \to \mathbb{R} \cup \{+\infty\}$ by defining f as $+\infty$ outside Q. Conversely, every extended real-valued convex function $f \colon \mathbb{E} \to \mathbb{R} \cup \{+\infty\}$ can be treated as a usual real-valued convex function $f \colon \operatorname{dom} f \to \mathbb{R}$ with the domain $\operatorname{dom} f$.

2.2.2 Strongly Convex Functions

Very often, we need to additionally require that a convex function is sufficiently curved and quantify somehow its curvature. The most common way to do this is by using the following definition.

Definition 2.2.4 (Strongly convex function)**.** Let $f \colon Q \to \mathbb{R}$ be a function, defined on a convex set $Q \subseteq \mathbb{E}$, and let $\|\cdot\|$ be a norm in $\mathbb{E}$. The function f

is called *strongly convex* (on Q) with constant $\mu > 0$ (w.r.t. the norm $\|\cdot\|$) if, for any $x, y \in Q$ and any $\alpha \in [0, 1]$, we have

$$f\big((1-\alpha)x + \alpha y\big) \leq (1-\alpha)f(x) + \alpha f(y) - \frac{\mu}{2}\alpha(1-\alpha)\|x-y\|^2.$$

Note that any strongly convex function is necessarily strictly convex and, in particular, convex. Also, the property of a function being strongly convex is independent of the particular choice of the norm, as any two norms in a finite-dimensional space are equivalent. However, the *constant* of strong convexity depends on the norm.

Similarly to Proposition 2.2.3, for differentiable functions, we have equivalent characterizations of strong convexity in terms of first and second derivatives (see Definition 2.1.3 and Theorems 2.1.9 and 2.1.11 in [133]).

Proposition 2.2.5. *Let $f\colon Q \to \mathbb{R}$ be a function, where $Q \subseteq \mathbb{E}$ is an open convex set, and let $\|\cdot\|$ be a norm in $\mathbb{E}$.*

(i) *Suppose f is differentiable on Q. Then, f is strongly convex on Q with constant $\mu > 0$ (w.r.t. $\|\cdot\|$) iff, for any $x, y \in Q$, we have*

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

(ii) *Suppose f is differentiable on Q. Then, f is strongly convex on Q with constant $\mu > 0$ (w.r.t. $\|\cdot\|$) iff, for any $x, y \in Q$, we have*

$$\langle \nabla f(x) - \nabla f(y), x - y\rangle \geq \mu\|x-y\|^2.$$

(iii) *Suppose f is twice differentiable on Q. Then, f is strongly convex on Q with constant $\mu > 0$ (w.r.t. $\|\cdot\|$) iff, for any $x \in Q$ and any $h \in \mathbb{E}$, we have*

$$\langle \nabla^2 f(x)h, h\rangle \geq \mu\|h\|^2.$$

In particular, if $\|\cdot\| := \|\cdot\|_B$ is the Euclidean norm, induced by some operator $B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$, then f is strongly convex on Q with constant $\mu > 0$ (w.r.t. $\|\cdot\|$) iff, for all $x \in Q$, we have

$$\nabla^2 f(x) \succeq \mu B. \tag{2.2.1}$$

Sometimes, the following result is useful (see Theorem 2.1.10 in [133]).

Proposition 2.2.6. *Let $f\colon Q \to \mathbb{R}$ be a function which is differentiable on an open convex set $Q \subseteq \mathbb{E}$, and let $\|\cdot\|$ be a norm in $\mathbb{E}$. Suppose that f is strongly convex with constant $\mu > 0$ (w.r.t. $\|\cdot\|$). Then, for all $x, y \in Q$,*

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2\mu} \|\nabla f(y) - \nabla f(x)\|_*^2.$$

2.2.3 Smooth Functions

We will also need some smoothness assumptions. Usually, they are expressed in terms of the Lipschitz continuity of the function and/or its derivatives.

In what follows, we make a convenient definition that $D^0 f \equiv f$ and $\|t\| \coloneqq |t|$ for any $t \in \mathbb{R}$.

Definition 2.2.7 (Lipschitz continuous derivatives)**.** Let $f\colon Q \to \mathbb{R}$ be a function, defined on an open convex set $Q \subseteq \mathbb{E}$, let $\|\cdot\|$ be a norm in $\mathbb{E}$, and let $p \geq 0$ be an integer. The function f is said to have *Lipschitz continuous derivative of order p* (on Q) with constant $L_p \geq 0$ (w.r.t. $\|\cdot\|$) if f is p times differentiable on Q, and, for any $x, y \in Q$, it holds[9]

$$\|D^p f(x) - D^p f(y)\| \leq L_p \|x - y\|. \tag{2.2.2}$$

Depending on the particular value of p in Definition 2.2.7, we use the following terminology. When $p = 0$, we say that f is a *Lipschitz continuous function*. If $p = 1$, we call f a function with *Lipschitz continuous gradient*. The case $p = 2$ corresponds to a function with *Lipschitz continuous Hessian*.

Similarly to strong convexity, the property of a function to have Lipschitz continuous derivative of a certain order does not depend on the particular choice of the norm. However, the corresponding Lipschitz *constant* does.

The Lipschitz continuity of a certain derivative can be equivalently characterized as the uniform boundedness of its next derivative.

Proposition 2.2.8. *Let $f\colon Q \to \mathbb{R}$ be a function which is $p+1$ times differentiable on an open convex set $Q \subseteq \mathbb{E}$ for some integer $p \geq 0$, and let $\|\cdot\|$ be a norm in $\mathbb{E}$. Then, the pth derivative of f is Lipschitz continuous on Q with constant $L_p \geq 0$ (w.r.t. $\|\cdot\|$) iff, for all $x \in Q$, we have*

$$\|D^{p+1} f(x)\| \leq L_p. \tag{2.2.3}$$

[9] Recall from the discussion in Sections 2.1.4 and 2.1.5 that $D^p f(x) - D^p f(y) \in \mathcal{L}_p$, where $\mathcal{L}_p$ is the space of linear operators defined in (2.1.15). Therefore, the norm in the left-hand side of (2.2.2) is the operator norm (2.1.18). In particular, for $p = 1$, this is exactly the dual norm (2.1.10).

In particular, if the norm $\|\cdot\|$ is Euclidean, then the pth derivative of f is Lipschitz continuous on Q with constant $L_p \geq 0$ (w.r.t. $\|\cdot\|$) iff, for all $x \in Q$ and all $h \in \mathbb{E}$, it holds

$$|D^{p+1} f(x)[h]^{p+1}| \leq L_p \|h\|^{p+1}.$$

Proof. i. Let us prove that (2.2.2) $\implies$ (2.2.3). Let $x \in Q$ and $h \in \mathbb{E}$ be arbitrary such that $\|h\| = 1$. Since the set Q is open, there exists $\bar{\tau} > 0$ such that $x + \tau h \in Q$ for all $\tau \in (0, \bar{\tau})$. According to (2.2.2), for all such τ,

$$\|D^p f(x + \tau h) - D^p f(x)\| \leq L_p \|\tau h\| = L_p \tau.$$

Dividing both sides by τ and passing to the limit as $\tau \to 0$ (taking into account the definition of the derivative (2.1.6)), we obtain

$$\|D^{p+1} f(x)[h]\| \leq L_p.$$

In view of (2.1.16), this proves (2.2.3) since the unit vector $h \in \mathbb{E}$ and the point $x \in Q$ were arbitrary.

ii. Now let us show that (2.2.3) $\implies$ (2.2.2). Let $x, y \in Q$ be arbitrary. By the fundamental theorem of calculus, we have

$$D^p f(y) - D^p f(x) = \int_0^1 D^{p+1} f(x + \tau(y - x))[y - x] d\tau. \tag{2.2.4}$$

Hence, by the triangle inequality for integrals, (2.1.17) and (2.2.3),

$$\begin{aligned} \|D^p f(y) - D^p f(x)\| &\leq \int_0^1 \|D^{p+1} f(x + \tau(y - x))[y - x]\| d\tau \\ &\leq \|y - x\| \int_0^1 \|D^{p+1} f(x + \tau(y - x))\| d\tau \\ &\leq L_p \|y - x\|. \end{aligned}$$

This proves (2.2.2) since $x, y \in Q$ were arbitrary.

iii. The second part of the claim (about the Euclidean norm) follows from the first one using (2.1.19) and the homogeneity of $D^{p+1} f(x)[\cdot]^{p+1}$. □

Let us also present Proposition 2.2.8 in an equivalent but more explicit form for the special case $p = 1$ which corresponds to the Lipschitz continuous gradient. For this, we can use the definition of the operator norm (2.1.18) and (2.1.9).

Corollary 2.2.9. *Let $f\colon Q \to \mathbb{R}$ be a twice differentiable function on an open convex set $Q \subseteq \mathbb{E}$, and let $\|\cdot\|$ be a norm in $\mathbb{E}$. Then, the gradient of f is Lipschitz continuous on Q with constant $L \geq 0$ (w.r.t. $\|\cdot\|$) iff, for all $x \in Q$ and all $h_1, h_2 \in \mathbb{E}$, we have*

$$\langle \nabla^2 f(x)h_1, h_2\rangle \leq L\|h_1\|\|h_2\|.$$

In particular, if $\|\cdot\| \coloneqq \|\cdot\|_B$ is the Euclidean norm, induced by some operator $B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$, then the gradient of f is Lipschitz continuous on Q with constant $L \geq 0$ (w.r.t. $\|\cdot\|$) iff, for all $x \in Q$, it holds

$$-LB \preceq \nabla^2 f(x) \preceq LB.$$

The following result provides a useful bound on the quality of the first-order Taylor approximation of the derivative of a smooth function.

Proposition 2.2.10. *Let $f\colon Q \to \mathbb{R}$ be a function defined on an open convex set $Q \subseteq \mathbb{E}$, let $\|\cdot\|$ be a norm in $\mathbb{E}$, and let $p \geq 0$ be an integer. Suppose that f has Lipschitz continuous derivative of order $p+1$ (on Q) with constant $L_{p+1} \geq 0$ (w.r.t. $\|\cdot\|$). Then, for all $x, y \in Q$, we have*

$$\|D^p f(y) - D^p f(x) - D^{p+1} f(x)[y-x]\| \leq \frac{L_{p+1}}{2}\|y-x\|^2.$$

Proof. Let $x, y \in Q$ be arbitrary. By the fundamental theorem of calculus,

$$\begin{aligned} &D^p f(y) - D^p f(x) - D^{p+1} f(x)[y-x] \\ &\quad = \int_0^1 \big(D^{p+1} f(x + \tau(y-x)) - D^{p+1} f(x)\big)[y-x]d\tau. \end{aligned}$$

Applying now the triangle inequality for integrals and using (2.1.17) and Lipschitz continuity of $D^{p+1}f$, we get

$$\begin{aligned} &\|D^p f(y) - D^p f(x) - D^{p+1} f(x)[y-x]\| \\ &\quad \leq \int_0^1 \|\big(D^{p+1} f(x + \tau(y-x)) - D^{p+1} f(x)\big)[y-x]\|d\tau \\ &\quad \leq \|y-x\| \int_0^1 \|D^{p+1} f(x + \tau(y-x)) - D^{p+1} f(x)\|d\tau \\ &\quad \leq L_{p+1}\|y-x\|^2 \int_0^1 td\tau = \frac{1}{2}L_{p+1}\|y-x\|^2. \end{aligned}$$

Putting everything together, we obtain the claim. □

2.2.4 Nonsmooth Convex Functions

Sometimes, we also need to work with convex functions which are not differentiable at certain points. For such functions, there exists a standard notion of a generalized derivative (cf. Proposition 2.2.3(i)).

Definition 2.2.11 (Subgradient)**.** Let $f\colon \mathbb{E} \to \mathbb{R}\cup\{+\infty\}$ be a convex function, and let $x \in \operatorname{dom} f$ be a point. A vector $g \in \mathbb{E}^*$ is called a *subgradient* of f at the point x if, for all $y \in \operatorname{dom} f$, we have

$$f(y) \geq f(x) + \langle g, y - x\rangle.$$

The set of all possible subgradients of the function f at the point x is called the *subdifferential* of f at x, and is denoted by $\partial f(x)$.

From the definition, it readily follows that, for any $x \in \operatorname{dom} f$, the subdifferential $\partial f(x)$ is a *closed convex* set (as the intersection of a certain collection of closed half-spaces). In principle, it may happen that $\partial f(x) = \varnothing$ for some $x \in \operatorname{dom} f$ (e.g., look at $\partial f(0)$ for the function $f\colon \mathbb{R} \to \mathbb{R}\cup\{+\infty\}$ defined by $f(x) \coloneqq -\sqrt{x}$ whenever $x \geq 0$, and $f(x) \coloneqq +\infty$ whenever $x < 0$). However, such pathological situations can only occur at the boundary of the effective domain: one of the basic results in Convex Analysis states that, for any $x \in \operatorname{int}\operatorname{dom} f$, the set $\partial f(x)$ is nonempty and bounded (see, e.g., Theorem 3.1.15 in [133]).

If f is differentiable at a point $x \in \operatorname{int}\operatorname{dom} f$, then $\partial f(x) = \{\nabla f(x)\}$, i.e., the gradient $\nabla f(x)$ is a unique subgradient at x. Conversely, if, at a point $x \in \operatorname{int}\operatorname{dom} f$, the subdifferential $\partial f(x)$ is a singleton, then f is differentiable at x, and the unique element of $\partial f(x)$ is exactly the gradient $\nabla f(x)$ (see Theorem 25.1 in [158]).

For nonsmooth convex functions, we have the following counterpart of Proposition 2.2.8 for the case $p = 0$.

Proposition 2.2.12. *Let $f\colon \mathbb{E} \to \mathbb{R}\cup\{+\infty\}$ be a convex function, let $Q \subseteq \operatorname{int}\operatorname{dom} f$ be an open convex set, and let $\|\cdot\|$ be a norm in $\mathbb{E}$. Then, f is Lipschitz continuous on Q with constant $M \geq 0$ (w.r.t. $\|\cdot\|$) iff, for all $x \in Q$ and all $f'(x) \in \partial f(x)$, we have*

$$\|f'(x)\|_* \leq M. \tag{2.2.5}$$

Proof. i. Let us prove that Lipschitz continuity implies (2.2.5). Let $x \in Q$, $h \in \mathbb{E}$ and $f'(x) \in \partial f(x)$ be arbitrary such that $\|h\| = 1$. Since the set Q is open, there exists $\tau > 0$ such that $x + \tau h \in Q$. By the definition of subgradient and Lipschitz continuity of f, we have

$$\langle f'(x), \tau h \rangle \leq f(x + \tau h) - f(x) \leq M\|\tau h\| = M\tau.$$

Dividing this inequality by τ, we get $\langle f'(x), h \rangle \leq M$. According to (2.1.10), this proves (2.2.5) since the unit vector $h \in \mathbb{E}$ and the point $x \in Q$ were arbitrary.

ii. Now let us show that (2.2.5) implies Lipschitz continuity. Let $x, y \in Q$ be arbitrary. Take[10] an arbitrary $f'(x) \in \partial f(x)$. Then, by the definition of subgradient, (2.1.11) and (2.2.5), we have

$$f(x) - f(y) \leq \langle f'(x), x - y \rangle \leq \|f'(x)\|_* \|x - y\| \leq M\|x - y\|.$$

Since $x, y \in Q$ were arbitrary, we can interchange them and obtain the corresponding reverse inequality. This proves that f is Lipschitz continuous on Q with constant M. □

2.3 Gradient Method

Consider the following unconstrained optimization problem:

$$\min_{x \in \mathbb{E}} f(x), \tag{2.3.1}$$

where $f : \mathbb{E} \to \mathbb{R}$ is a differentiable function.

We assume that, the objective function in problem (2.3.1) is strongly convex with some constant $\mu > 0$ and its gradient is Lipschitz continuous with some constant $L > 0$, i.e., for all $x, y \in \mathbb{E}$, we have (see Proposition 2.2.5(ii) and Definition 2.2.7)

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \mu\|x - y\|^2, \tag{2.3.2}$$

$$\|\nabla f(x) - \nabla f(y)\|_* \leq L\|x - y\|, \tag{2.3.3}$$

where $\|\cdot\|$ is a Euclidean norm, generated by some $B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$:

$$\|x\| \coloneqq \|x\|_B \coloneqq \langle Bx, x \rangle^{1/2}. \tag{2.3.4}$$

[10] Note that $\partial f(x) \neq \varnothing$ since $x \in \operatorname{int} Q$.

An important characteristic of such a function is its *condition number*:

$$\varkappa := \frac{L}{\mu} \geq 1. \tag{2.3.5}$$

Note that, in view of our assumptions, a solution of problem (2.3.1) exists and is unique. Let us denote the corresponding optimal value by f^*.

Consider the simplest Gradient Method with constant step size for solving problem (2.3.1).

Algorithm 2.3.1: Gradient Method with Constant Step Size
Input: Initial point $x_0 \in \mathbb{E}$.
Iteration $k \geq 0$**:** Compute the new point $x_{k+1} := x_k - \frac{1}{L} B^{-1} \nabla f(x_k)$.

In this method, we assume that the Lipschitz constant L is known. The operator B can be thought of as a certain "preconditioner". By choosing it appropriately, we may improve the condition number (2.3.5) of the function f. However, at the same time, B should be sufficiently simple so that we can efficiently compute $B^{-1}\nabla f(x_k)$ at every iteration.

Let us present a standard efficiency bound for Algorithm 2.3.1.

Theorem 2.3.1. *In Algorithm 2.3.1, for all $k \geq 0$, we have*

$$f(x_k) - f^* \leq (1 - \varkappa^{-1})^k [f(x_0) - f^*]. \tag{2.3.6}$$

Proof. Let $k \geq 0$ be arbitrary. Applying Proposition 2.2.10 and using the definition of x_{k+1} from Algorithm 2.3.1, we obtain

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{L}{2} \|x_{k+1} - x_k\|^2 \\ &= f(x_k) - \frac{1}{2L} \|\nabla f(x_k)\|_*^2. \end{aligned}$$

At the same time, since $\nabla f(x^*) = 0$, by Proposition 2.2.6, we have

$$f(x_k) - f^* \leq \frac{1}{2\mu} \|\nabla f(x)\|_*^2.$$

Combining the above two inequalities and using (2.3.5), we obtain

$$f(x_k) - f(x_{k+1}) \geq \frac{1}{2L}\|\nabla f(x_k)\|_*^2 \geq \varkappa^{-1}[f(x_k) - f^*].$$

Thus,

$$f(x_{k+1}) - f^* \leq (1 - \varkappa^{-1})[f(x_k) - f^*],$$

and (2.3.6) follows since $k \geq 0$ was arbitrary. □

According to Theorem 2.3.1, Algorithm 2.3.1 has a global linear rate of convergence with constant depending only on the condition number of the problem. From (2.3.6), we can easily estimate the number of iterations to obtain a point $\bar{x} \in \mathbb{E}$ such that $f(\bar{x}) - f^* \leq \varepsilon$ for some $0 < \varepsilon \leq f(x_0) - f^*$:

$$\varkappa \ln \frac{f(x_0) - f^*}{\varepsilon}. \tag{2.3.7}$$

The main factor in this complexity bound is the condition number $\varkappa$.

In practice, instead of using the constant step size $1/L$, it might be better to use a certain adaptive *line search* procedure which tunes the step size automatically at each iteration. This is useful for two reasons. First, the actual value of L may be unknown or difficult to estimate. Second, even if L is known, it might still better to use line search because, for some iterations, it may find better *local* estimates of the Lipschitz constant L and thus make bigger steps.

Let us present a version of the Gradient Method with line search. As input, it takes some initial point x_0 and some initial estimate $\tilde{L}_0$ of the actual Lipschitz constant L.

Algorithm 2.3.2: Gradient Method with Line Search

Input: Initial point $x_0 \in \mathbb{E}$ and Lipschitz estimate $\tilde{L}_0 \in (0, L]$.

Iteration $k \geq 0$**:**

1. Set $L_{k,0} := \tilde{L}_k$.
2. Iterate for $i \geq 0$:
   a) Compute the new trial point
   $$x_{k+1,i} := x_k - \frac{1}{L_{k,i}} B^{-1} \nabla f(x_k).$$
   b) Check if the trial point is good enough:
   $$f(x_k) - f(x_{k+1,i}) \geq \frac{1}{2L_{k,i}} \|\nabla f(x_k)\|_*^2.$$
   If yes, set $i_k := i$ and break the loop.
   c) Set $L_{k,i+1} := 2L_{k,i}$.
3. Set $x_{k+1} := x_{k+1,i_k}$, $\tilde{L}_{k+1} := L_{k,i_k}/2$.

Note that, in contrast to an upper bound on the actual Lipschitz constant L, a lower bound $\tilde{L}_0 > 0$ on L can be easily computed. For example, one can take any point $x_0' \in \mathbb{E}$, different from x_0, and set

$$\tilde{L}_0 := \frac{\|\nabla f(x_0') - \nabla f(x_0)\|_*}{\|x_0' - x_0\|}.$$

Then, $\tilde{L}_0 \in (0, L]$ in view of (2.3.3) and (2.3.2).

Note also that the inner loop (Step 2) in Algorithm 2.3.2 is always finite. Indeed, in the worst case, at some moment, the estimate $L_{k,i}$ will become greater or equal than the actual Lipschitz constant L, and so the condition at Step 2b will be satisfied in view of Proposition 2.2.10.

Theorem 2.3.2. *For all $k \geq 0$, in Algorithm 2.3.2, we have*

$$f(x_k) - f^* \leq \left(1 - (2\varkappa)^{-1}\right)[f(x_k) - f^*]. \tag{2.3.8}$$

Moreover, for any $k \geq 1$, the total number of line search iterations during the course of the first k iterations of Algorithm 2.3.2 is

$$\sum_{t=0}^{k-1} (i_t + 1) \leq 2k + \log_2(L/\tilde{L}_0). \tag{2.3.9}$$

Proof. First, let us show, by induction, that, for all $k \geq 0$, we have

$$\tilde{L}_k \leq L. \tag{2.3.10}$$

Clearly, (2.3.10) is satisfied when $k = 0$ in view of the assumption on $\tilde{L}_0$ in Algorithm 2.3.2. Now suppose that (2.3.10) has been proved for all indices from 0 up to some $k \geq 0$. In view of Proposition 2.2.10, the condition at Step 2b is definitely satisfied once $L_{k,i} \equiv 2^i \tilde{L}_k$ becomes greater or equal than L. Combining this observation with the fact that $\tilde{L}_k \leq L$ and $i_k \geq 0$ is the first integer for which the condition at Step 2b is satisfied, we obtain $L_{k,i_k} \leq 2L$. Therefore, $\tilde{L}_{k+1} \equiv L_{k,i_k}/2 \leq L$, which proves (2.3.10) for the next index $k + 1$. Thus, (2.3.10) is now proved for all indices.

Let $k \geq 0$ be arbitrary. From (2.3.10), it follows that

$$L_{k,i_k} \equiv 2\tilde{L}_{k+1} \leq 2L.$$

Combining this with the condition at Step 2b, we obtain

$$f(x_k) - f(x_{k+1}) \geq \frac{1}{2L_{k,i_k}} \|\nabla f(x_k)\|_*^2 \geq \frac{1}{4L} \|\nabla f(x_k)\|_*^2.$$

On the other hand, since $\nabla f(x^*) = 0$, we have $f(x_k) - f^* \leq \frac{1}{2\mu} \|\nabla f(x_k)\|_*^2$ by strong convexity (see Proposition 2.2.6). Thus,

$$f(x_k) - f(x_{k+1}) \geq (2\varkappa)^{-1} [f(x_k) - f^*],$$

and (2.3.8) follows.

It remains to prove (2.3.9). Note that, for any $k \geq 0$, we have

$$\tilde{L}_{k+1} = L_{k,i_k}/2 = (2^{i_k} \tilde{L}_k)/2 = 2^{i_k - 1} \tilde{L}_k.$$

Therefore, for all $k \geq 0$,

$$i_k - 1 = \log_2(\tilde{L}_{k+1}/\tilde{L}_k).$$

Consequently, for all $k \geq 1$,

$$\sum_{t=0}^{k-1} (i_t + 1) = 2k + \log_2(\tilde{L}_k/\tilde{L}_0).$$

Applying (2.3.10), we obtain (2.3.9). □

Theorem 2.3.2 shows that the worst-case efficiency estimate of the Gradient Method with line search from Algorithm 2.3.2 is the same (up to an absolute constant) as that of the basic method from Algorithm 2.3.1, which uses the constant step size $1/L$. Moreover, the line search does not add any significant overhead since the *average* number of auxiliary line search iterations (spent inside each "outer" iteration k of the method) quickly approaches the constant 2. In practice, however, in the majority of cases, the Gradient Method with line search is much faster than that with constant step size.

2.4 Newton's Method

A natural idea to accelerate the Gradient Method is to additionally use the second derivatives of the objective function. This leads us to Newton's Method.

In this section, we consider the following unconstrained optimization problem:

$$\min_{x \in \mathbb{E}} f(x), \tag{2.4.1}$$

where $f : \mathbb{E} \to \mathbb{R}$ is a twice differentiable convex function. We assume that problem (2.4.1) has a solution and denote the corresponding optimal value by f^*.

2.4.1 Classical Newton's Method

Newton's Method is based on the simple idea of approximating the function with its quadratic Taylor model around the current iterate x_k,

$$f(x) \approx f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2} \langle \nabla^2 f(x_k)(x - x_k), x - x_k \rangle,$$

and then choosing the next point x_{k+1} as a minimizer of this model. In order for this minimizer to exist and be unique, one should require that the Hessian of f is strictly positive definite. With this assumption, x_{k+1} is well-defined and can be computed in the closed form.

Algorithm 2.4.1: Newton's Method
Input: Initial point $x_0 \in \mathbb{E}$.
Iteration $k \geq 0$**:** Compute the new point $x_{k+1} := x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$.

Comparing Algorithm 2.4.1 with Algorithm 2.3.1, we see that Newton's Method can be considered a variant of the Gradient Method in which the fixed "preconditioning" operator B is replaced with the Hessian at the current point and the step size $1/L$ equals 1.

Let us present standard efficiency estimates for Newton's Method. For this, we need to introduce additional regularity assumptions about the objective function f in problem (2.4.1).

Standard Analysis

The classical assumptions for the analysis of Newton's Method are as follows: f is strongly convex with constant $\mu > 0$ and its Hessian is Lipschitz continuous with constant[11] $L_2 > 0$, i.e., for all $x, y \in \mathbb{E}$, we have

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \mu \|x - y\|^2, \tag{2.4.2}$$

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L_2 \|x - y\|, \tag{2.4.3}$$

where $\|\cdot\|$ is some Euclidean norm in $\mathbb{E}$.

Under these assumptions, we can easily establish the following key inequality for Newton's Method.

Lemma 2.4.1. *In Algorithm 2.4.1, for all $k \geq 0$, we have*

$$\|\nabla f(x_{k+1})\|_* \leq \frac{L_2}{2\mu^2} \|\nabla f(x_k)\|_*^2.$$

Proof. Let $k \geq 0$ be arbitrary. By the definition of x_{k+1}, we have

$$\nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1} - x_k) = 0.$$

[11] The case $L_2 = 0$ is not really interesting since then the function f is quadratic and Newton's Method finds an exact solution of problem (2.4.1) after one step.

Combining this with Proposition 2.2.10, we obtain

$$\begin{aligned}\|\nabla f(x_{k+1})\|_* &= \|\nabla f(x_{k+1}) - \nabla f(x_k) - \nabla^2 f(x_{k+1})(x_{k+1} - x_k)\|_* \\ &\leq \frac{L_2}{2}\|x_{k+1} - x_k\|^2.\end{aligned}$$

It remains to note that, by strong convexity, we have

$$\|x_{k+1} - x_k\| = \|[\nabla^2 f(x_k)]^{-1}\nabla f(x_k)\| \leq \frac{1}{\mu}\|\nabla f(x_k)\|_*.$$

Indeed, if $B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ is the operator, defining $\|\cdot\|$, then, by (2.2.1), we have $\nabla^2 f(x_k) \succeq \mu B$, and so $[\nabla^2 f(x_k)]^{-1} B [\nabla^2 f(x_k)]^{-1} \preceq \mu^{-2} B^{-1}$. □

From Lemma 2.4.1, it follows that Newton's Method has *local quadratic convergence*.

Theorem 2.4.2. *Suppose that, in Algorithm 2.4.1, the initial point x_0 is sufficiently good:*

$$\frac{L_2}{2\mu^2}\|\nabla f(x_0)\|_* \leq \frac{1}{2}. \tag{2.4.4}$$

Then, for all $k \geq 0$, we have

$$\frac{L_2}{2\mu^2}\|\nabla f(x_k)\|_* \leq 2^{-2^k}. \tag{2.4.5}$$

Proof. Denote $M := L_2/(2\mu^2)$ and $g_k := \|\nabla f(x_k)\|_*$ for all $k \geq 0$. According to Lemma 2.4.1, for all $k \geq 0$, we have

$$M g_{k+1} \leq (M g_k)^2.$$

Unrolling this recurrence, we obtain, for all $k \geq 0$,

$$M g_k \leq (M g_0)^{2^k},$$

and (2.4.5) follows since $M g_0 \leq \frac{1}{2}$ in view (2.4.4). □

Theorem 2.4.2 gives us the rate of the convergence of Newton's Method in terms of the norm of the gradient. However, from this result, we can easily obtain the corresponding rate for function values. Indeed, applying

Proposition 2.2.6 and (2.4.5), we obtain, for all $k \geq 0$,

$$f(x_k) - f^* \leq \frac{1}{2\mu}\|\nabla f(x_k)\|_*^2 \leq \frac{1}{2\mu}\Big(\frac{2\mu^2}{L_2}2^{-2^k}\Big)^2 = \frac{2\mu^3}{L_2^2}2^{-2^{k+1}}.$$

Consequently, to find a point $\bar{x} \in \mathbb{E}$ such that $f(\bar{x}) - f^* \leq \varepsilon$, Newton's Method requires at most the following number of iterations:

$$\log_2 \log_2 \frac{2\mu^3}{L_2^2 \varepsilon} \tag{2.4.6}$$

(assuming that $0 < \varepsilon < 2\mu^3/L_2^2$). We see that the accuracy ε and all the parameters of the problem class enter this estimate under the *double logarithm*. This is an extremely fast convergence rate. Nevertheless, it is important to remember that complexity bound (2.4.6) is valid only under the assumption that the initial point x_0 in Newton's Method is sufficiently good, as specified in (2.4.4).

Self-Concordant Analysis

Despite the apparent naturalness and simplicity of the standard local convergence analysis of Newton's Method, presented above, it has one hidden flaw. Observe that the final complexity estimate (2.4.6) for Newton's Method, as well as its region of local convergence, described in (2.4.4), are expressed in terms of a certain Euclidean norm $\|\cdot\|$. In particular, both constants μ and L_2 depend on the norm $\|\cdot\|$ (see (2.4.2) and (2.4.3)). At the same time, Newton's Method (Algorithm 2.4.1) itself does not depend on the choice of the norm $\|\cdot\|$. Thus, we have a very strange situation when the analysis of a method is written in terms of some norm which has nothing to do with the method itself. Note that, for the Gradient Method (Algorithm 2.3.1), there is no such problem since the method explicitly depends on the operator B, which defines the norm $\|\cdot\|$.

The aforementioned deficiency in the standard analysis of Newton's Method was first noticed and addressed by Nesterov and Nemirovski [139]. Instead of measuring derivatives of the objective function f is some arbitrary Euclidean norm, they proposed measuring them in the Euclidean norm, induced by the function itself. This idea lead them to the definition of *self-concordant functions*.

Definition 2.4.3 (Self-concordant function)**.** Let $f\colon \mathbb{E} \to \mathbb{R} \cup \{+\infty\}$ be a

closed[12] convex function with open effective domain $\operatorname{dom} f$. Suppose that f is three times differentiable on $\operatorname{dom} f$ and its Hessian is nondegenerate on $\operatorname{dom} f$. Then, f is called *self-concordant* with constant $M \geq 0$ if, for all $x \in \operatorname{dom} f$ and all $h \in \mathbb{E}$, we have

$$D^3 f(x)[h]^3 \leq 2M \|h\|_x^3, \tag{2.4.7}$$

where $\|h\|_x := \|h\|_{\nabla^2 f(x)}$.

Comparing inequality (2.4.7) with a similar inequality from Proposition 2.2.8 for functions with Lipschitz continuous Hessian, we see that self-concordance can be viewed as *local* Lipschitz continuity of the Hessian, measured w.r.t. the local Euclidean norm $\|\cdot\|_x$. The constant $2M$ plays the role of the local Lipschitz constant of the Hessian. Note that there is no need to define the similar version of local strong convexity since it is satisfied automatically with constant 1: for any $x \in \operatorname{dom} f$ and any $h \in \mathbb{E}$, we have $\langle \nabla^2 f(x)h, h\rangle \equiv \|h\|_x^2$ (cf. Proposition 2.2.5(iii)).

The simplest and most important example of a self-concordant function is the negative logarithm: $f(x) := -\ln x$ with $\operatorname{dom} f := (0, +\infty)$. More generally, it is known that many standard operations on convex functions preserve self-concordance: weighted sum, composition with affine mapping, partial minimization, etc. For more details and other examples of self-concordant functions, see, e.g., Chapter 5 in [133].

Another interesting example of a self-concordant function is given by a strongly convex function with Lipschitz continuous Hessian.

Lemma 2.4.4. *Let $f : \mathbb{E} \to \mathbb{R}$ be a three times differentiable function. Suppose that f is strongly convex with constant $\mu > 0$ and its Hessian is Lipschitz continuous with constant $L_2 \geq 0$ (w.r.t. to a certain norm $\|\cdot\|$). Then, f is self-concordant with constant*

$$M := \frac{L_2}{2\mu^{3/2}}.$$

[12] Recall that the function $f : \mathbb{E} \to \mathbb{R} \cup \{+\infty\}$ is called *closed* if its epigraph $\operatorname{epi} f := \{(x, t) \in \mathbb{E} \times \mathbb{R} : f(x) \leq t\}$ is a closed set. It is not difficult to see that, in the case when $\operatorname{dom} f$ is open and f is continuous on $\operatorname{dom} f$, the closedness of f is actually equivalent to the *barrier property*: for any $\bar{x} \in \partial(\operatorname{dom} f)$, it holds that $f(x) \to +\infty$ as $x \to \bar{x}; x \in \operatorname{dom} f$, where $\partial(\operatorname{dom} f)$ is the boundary of $\operatorname{dom} f$.

Proof. Indeed, applying Propositions 2.2.8 and 2.2.5(iii), we obtain

$$D^3 f(x)[h]^3 \le L_2 \|h\|^3 \le \frac{L_2}{\mu^{3/2}} \|h\|_x^3$$

since $\|h\|_x^2 \equiv \langle \nabla^2 f(x)h, h\rangle \ge \mu \|h\|^2$ by Proposition 2.2.5(iii). □

Thus, self-concordant functions form a bigger class than strongly convex functions with Lipschitz continuous Hessian, which we studied earlier.

Let us present a local convergence analysis for Newton's Method (Algorithm 2.4.1) for minimizing a self-concordant function $f : \mathbb{E} \to \mathbb{R} \cup \{+\infty\}$ with parameter $M > 0$.

Similarly to our previous analysis, it will be convenient to measure the progress of Newton's Method in terms of the norm of the gradient. However, now we will use a local norm:

$$\lambda_f(x) := \|\nabla f(x)\|_x^*, \qquad x \in \operatorname{dom} f.$$

For self-concordant functions, we have the following key inequality for one step of Newton's Method (cf. Lemma 2.4.1).

Proposition 2.4.5 (see Theorem 5.2.2 in [133])**.** *Consider iteration $k \ge 0$ of Algorithm 2.4.1. Suppose that $x_k \in \operatorname{dom} f$ and $M\lambda_f(x_k) < 1$. Then, $x_{k+1} \in \operatorname{dom} f$ and*

$$\lambda_f(x_{k+1}) \le \frac{M\lambda_f^2(x_k)}{(1 - M\lambda_f(x_k))^2}.$$

From Proposition 2.4.5, we obtain the following local efficiency estimate for Newton's Method, applied for minimizing a self-concordant function.

Theorem 2.4.6. *Let the initial point $x_0 \in \operatorname{dom} f$ in Algorithm 2.4.1 be sufficiently good:*

$$M\lambda_f(x_0) \le \rho := 2 - \sqrt{3} \ (\approx 0.267\ldots). \tag{2.4.8}$$

Then, for all $k \ge 0$, we have $x_k \in \operatorname{dom} f$ and

$$M\lambda_f(x_k) \le (2\rho)2^{-2^k} \ (\le \rho). \tag{2.4.9}$$

Proof. Denote $\lambda_k := \lambda_f(x_k)$ for all $k \ge 0$. Let us prove by induction that (2.4.9) holds (with $x_k \in \operatorname{dom} f$) for all $k \ge 0$. When $k = 0$, this follows from our assumptions. Now suppose that we have already proved the inductive

hypothesis for all indices from 0 up to some $k \geq 0$. Then, from (2.4.9), it follows that $M\lambda_k \leq \rho < 1$. Applying Proposition 2.4.5 and (2.4.9), we obtain $x_{k+1} \in \operatorname{dom} f$ and

$$M\lambda_{k+1} \leq \frac{(M\lambda_k)^2}{(1-M\lambda_k)^2} \leq \frac{(M\lambda_k)^2}{(1-\rho)^2} \leq \Big(\frac{2\rho}{1-\rho}\Big)^2 2^{-2^{k+1}} = (2\rho)2^{-2^{k+1}}$$

since $(1-\rho)^2 = (\sqrt{3}-1)^2 = 2\rho$ by the definition of ρ in (2.4.8). This proves that (2.4.9) is also valid for the next index $k+1$. □

In order to obtain a more meaningful convergence guarantee for Newton's Method in terms of function value, we need to relate the local norm of the gradient with the functional residual. This can be done using the following result (see Theorem 5.2.1 and Lemma 5.1.5 from [133]).

Proposition 2.4.7. *Let $f\colon \mathbb{E} \to \mathbb{R} \cup \{+\infty\}$ be a self-concordant function with constant $M \geq 0$. Let $x \in \operatorname{dom} f$ be such that $M\lambda_f(x) < 1$, and let f^* be the minimal value of f. Then,*

$$f(x) - f^* \leq \frac{\lambda_f^2(x)}{2(1-M\lambda_f(x))}.$$

Combining Proposition 2.4.7 and Theorem 2.4.6, we obtain, for all $k \geq 0$,

$$M^2[f(x_k) - f^*] \leq \frac{(M\lambda_f(x_k))^2}{2(1-\rho)} \leq \frac{(2\rho)^2}{2(1-\rho)} 2^{-2^{k+1}} = \frac{2\rho^2}{1-\rho} 2^{-2^{k+1}}.$$

Thus, to find $\bar{x} \in \operatorname{dom} f$ such that $f(\bar{x}) - f^* \leq \varepsilon$, Newton's Method needs at most the following number of iterations:

$$\log_2 \log_2 O\Big(\frac{1}{M^2\varepsilon}\Big), \tag{2.4.10}$$

where $O(\cdot)$ hides a certain absolute constant, namely, $2\rho^2/(1-\rho) = 0.196\ldots$

Note that, at this point, we have two different local analyses of Newton's Method, applied for minimizing a μ-strongly convex function f with L_2-Lipschitz continuous Hessian. First, we can apply the "old" analysis from the previous section. Alternatively, according to Lemma 2.4.4, we can treat the function f as self-concordant with parameter

$$M = \frac{L_2}{2\mu^{3/2}},$$

and apply the "new" analysis for general self-concordant functions. Let us show that the latter approach is better. Indeed, according to the "old" analysis, the final complexity estimate for obtaining an ε-approximate solution in terms of the functional residual is as follows (see (2.4.6)):

$$\log_2 \log_2 O\Big(\frac{\mu^3}{L_2^2 \varepsilon}\Big) \tag{2.4.11}$$

The "old" description of the region of local convergence is (see (2.4.4)):

$$\|\nabla f(x_0)\|_* \leq O\Big(\frac{\mu^2}{L_2}\Big). \tag{2.4.12}$$

According to the "new" analysis, the final complexity estimate, given by (2.4.10), is exactly the same as the "old" one from (2.4.11) (up to an absolute constant). However, the "new" description of the region of local convergence is much better (see (2.4.8)):

$$\|\nabla f(x_0)\|_{x_0}^* \leq O\Big(\frac{\mu^{3/2}}{L_2}\Big). \tag{2.4.13}$$

Indeed, by strong convexity, we have

$$\|\nabla f(x_0)\|_{x_0}^* \equiv \langle \nabla f(x_0), [\nabla^2 f(x_0)]^{-1} \nabla f(x_0) \rangle^{1/2} \leq \frac{1}{\sqrt{\mu}} \|\nabla f(x_0)\|_*.$$

Therefore, any point x_0, belonging to the "old" region (2.4.12), also belongs to the "new" region (2.4.13), but not vice versa. In other words, the "new" region of convergence is larger than the "old" one. This is another confirmation that, for the analysis of the classical Newton's Method, it is better to work with the class of self-concordant functions and in terms of local norms.

2.4.2 Globally Convergent Variants

Unfortunately, the Classical Newton's Method is not *globally* convergent: it may fail to converge if the initial point is not sufficiently good (even under assumptions (2.4.2) and (2.4.3), see, e.g., Example 1.4.3 in [51]). However, there exist other variants of Newton's Method which are free of this flaw. Let us discuss two of them. Our presentation will be brief since, as we already explained in Section 1.1, in this thesis, our focus is mainly on *local* efficiency estimates.

Damped Newton's Method

The simplest strategy is to use the *Damped Newton's Method*:

$$x_{k+1} := x_k - h_k[\nabla^2 f(x_k)]^{-1}\nabla f(x_k), \qquad k \geq 0, \tag{2.4.14}$$

where $h_k > 0$ is a certain step size parameter, which can be tuned using line search. It is important that the line search eventually switches to the unit step size $h_k \equiv 1$ in order for method (2.4.14) to have local quadratic convergence.

Under the assumption that the objective function f is μ-strongly convex and has L-Lipschitz continuous gradient, one can prove that, in order to find an ε-approximate solution (in terms of function value) to problem (2.4.1), Damped Newton's Method (equipped with a certain line search strategy for choosing h_k) requires at most the following number of iterations:

$$O\Big(\varkappa^2 \ln \frac{f(x_0) - f^*}{\varepsilon}\Big), \tag{2.4.15}$$

where $\varkappa := L/\mu \geq 1$ is the condition number (see, e.g., Section 1.4.2 in [51]).

Comparing bound (2.4.15) with the corresponding bound (2.3.7) for the Gradient Method, we see that it is worse: now the complexity is proportional to $\varkappa^2$ instead of $\varkappa$. However, despite such a pessimistic theoretical conclusion, in practice, the Damped Newton's Method is much faster than the Gradient Method. This can be partly explained as follows. First, we should keep in mind that, in contrast to the Gradient Method, the Damped Newton's Method is *affine-invariant*[13]: it does not depend on the particular norm $\|\cdot\|$, which is used for defining the constants μ and L. In other words, the condition number $\varkappa$ in complexity bound (2.4.15) can be taken w.r.t. an arbitrary norm, and therefore can potentially be much smaller than the corresponding condition number for the Gradient Method. Second, once the Damped Newton's Method reaches the region of local convergence of the Classical Newton's Method, it automatically accelerates to a quadratic convergence rate, which does not happen with the Gradient Method.

[13] Here we implicitly assume that the rule for choosing step sizes h_k in (2.4.14) is also affine-invariant, which is typically the case for any standard line search procedure (working with the local norm $\|\cdot\|_{x_k}$).

Cubic Newton's Method

Another approach for globalizing Newton's Method is based on the idea of *Cubic Regularization* of the second-order Taylor model, used in the Classical Newton's Method.

Let us introduce the following auxiliary function for each $x, y \in \mathbb{E}$ and each $H > 0$:

$$\hat{f}_H(x, y) \coloneqq f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2}\langle \nabla^2 f(x)(y - x), y - x \rangle + \frac{H}{6}\|y - x\|^3,$$

where $\|\cdot\|$ is a Euclidean norm. If f has L_2-Lipschitz continuous Hessian, then, for any $H \geq L_2$, this auxiliary function provides us with a *global upper bound* on the objective function f: for all $x, y \in \mathbb{E}$, we have

$$f(y) \leq \hat{f}_H(x, y).$$

The *Cubic Newton's Method* successively minimizes this upper bound:

$$x_{k+1} = \operatorname*{argmin}_{x \in \mathbb{E}} \hat{f}_{H_k}(x_k, x), \qquad k \geq 0, \tag{2.4.16}$$

where $H_k > 0$ are certain "step size" parameters, which can be automatically tuned using "line search" (similarly to Algorithm 2.3.2).

Iteration (2.4.16) corresponds to solving the following system of nonlinear equations:

$$\begin{aligned} x_{k+1} &= x_k - [\nabla^2 f(x_k) + \tfrac{1}{2} H_k r_k B]^{-1} \nabla f(x_k), \\ r_k &= \|x_{k+1} - x_k\|, \end{aligned} \tag{2.4.17}$$

where $B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ is the operator, defining the norm $\|\cdot\|$. In this form, the Cubic Newton's Method can be seen as a variant of the *Levenberg–Marquardt Method*,

$$x_{k+1} = x_k - [\nabla^2 f(x_k) + \lambda_k B]^{-1} \nabla f(x_k), \qquad k \geq 0,$$

with an implicit rule for choosing the regularization parameter λ_k.

Recently, it was proved that, for finding an ε-approximate solution in terms of the function value, the Cubic Newton's Method, applied for minimizing a μ-strongly convex function f with L-Lipschitz continuous gradient,

needs at most

$$O\Big(\varkappa \ln \frac{f(x_0) - f^*}{\varepsilon}\Big) \tag{2.4.18}$$

iterations, where $\varkappa := L/\mu$ is the condition number (see [54]).

Comparing bound (2.4.18) with the corresponding bound (2.3.7) for the Gradient Method, we see that they are identical. Thus, the Cubic Newton's Method is at least as fast as the Gradient Method[14]. In fact, it turns out to be strictly faster. Indeed, a more refined analysis from [54] reveals that, instead of the usual "first-order" condition number $\varkappa$ in (2.4.18), there should be a certain "second-order" condition number $\hat{\varkappa} \leq \varkappa$, which is insensitive to any quadratic parts of the objective function. More precisely, the addition of any convex quadratic function (even highly ill-conditioned) to the objective function can only improve $\hat{\varkappa}$, which is not the case for $\varkappa$. Furthermore, similarly to the Classical Newton's Method, the Cubic Newton's Method has local quadratic convergence.

2.5 Quasi-Newton Methods

We have seen that Newton's Method is much more efficient than the Gradient Method. However, each iteration of Newton's Method requires computing the Hessian and solving a linear system with it, which can be very expensive for large-scale problems. Quasi-Newton methods aim at approximating Newton's Method without the need for computing the Hessian.

In this section, we consider the same problem as before, namely,

$$\min_{x \in \mathbb{E}} f(x), \tag{2.5.1}$$

where $f\colon \mathbb{E} \to \mathbb{R}$ is a twice continuously differentiable function with strictly positive definite Hessian.

[14] Here we are speaking about *analytical complexities* (the number of oracle calls) of the two methods. Of course, the corresponding *arithmetical complexities* (the total number of arithmetical operations) might be quite different depending on a particular problem under consideration. Nevertheless, in many real-world problems of moderate dimension, a significant improvement in the analytical complexity often leads to an improvement in the arithmetical complexity as well.

2.5.1 General Scheme

Quasi-Newton methods are based on the following iteration:

$$x_{k+1} = x_k - h_k G_k^{-1} \nabla f(x_k), \qquad k \geq 0, \tag{2.5.2}$$

where $h_k \geq 0$ is a step size, and $G_k \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ is a positive definite linear operator, which approximates the Hessian at the current point:

$$G_k \approx \nabla^2 f(x_k).$$

The goal is to update G_k at each iteration to ensure that it becomes an increasingly accurate approximation of the actual Hessian. However, the cost of the corresponding update should be much lower than that of computing the exact Hessian.

Note that, in principle, for efficiently implementing iteration (2.5.2), we do not really need the Hessian approximations G_k themselves. What we actually need is their inverses:

$$H_k \coloneqq G_k^{-1} \approx [\nabla^2 f(x_k)]^{-1}.$$

Therefore, instead of updating G_k and then explicitly inverting it, it makes sense to directly update the inverse Hessian approximation H_k.

We thus come to the following general scheme of a quasi-Newton method.

Algorithm 2.5.1: General Scheme of a Quasi-Newton Method
Initialization: Choose $x_0 \in \mathbb{E}$ and $H_0 \in \mathcal{S}_{++}(\mathbb{E}^*, \mathbb{E})$.
Iteration $k \geq 0$**:** 1. Compute a step size $h_k \geq 0$. 2. Compute $x_{k+1} \coloneqq x_k - h_k H_k \nabla f(x_k)$. 3. Update H_k into H_{k+1}.

For computing the step size h_k at each iteration of Algorithm 2.5.1, there exist several standard approaches based on the *line search* in the direction

$$d_k \coloneqq H_k \nabla f(x_k). \tag{2.5.3}$$

Specifically, one attempts to find a sufficiently large step size $h_k \geq 0$, for

which the function value at the new point, $f(x_{k+1}) \equiv f(x_k - h_k d_k)$, is sufficiently smaller than that at the current point.

One particular line search strategy, which is especially popular in the context of quasi-Newton methods, prescribes selecting a step size $h_k \geq 0$ which satisfies the *Wolfe conditions*:

$$\begin{aligned} f(x_k - h_k d_k) &\leq f(x_k) - c_1 h_k \langle \nabla f(x_k), d_k \rangle, \\ \langle \nabla f(x_k - h_k d_k), d_k \rangle &\leq c_2 \langle \nabla f(x_k), d_k \rangle, \end{aligned} \tag{2.5.4}$$

where $0 < c_1 < c_2 < 1$ are certain parameters. In order to achieve local superlinear convergence, it is important to always try the unit step size $h_k \equiv 1$ first and accept it whenever it satisfies conditions (2.5.4). For more information about the Wolfe conditions and an efficient algorithm, which can be used for finding a step size, satisfying them, we refer the reader to Sections 3.1 and 3.5 in [144].

Another popular procedure, which can be used for computing the step size h_k in Algorithm 2.5.1, is the following *backtracking line search*.

Algorithm 2.5.2: Backtracking Line Search
Input: Constants $c_1 \in (0, 1)$ and $0 < \tau \leq \tau' < 1$.
1. Set $h := 1$.
2. Until $f(x_k - h d_k) \leq f(x_k) - c_1 h \langle \nabla f(x_k), d_k \rangle$ is not satisfied, select a new $h \in [\tau h, \tau' h]$.
3. Return $h_k := h$.

The simplest version of Algorithm 2.5.2 corresponds to choosing the new h at Step 2 by halving the previous one: $\tau := \tau' := 0.5$. A more advanced variant of this procedure chooses the new h by minimizing a certain interpolation polynomial for the function $h \to f(x_k - h d_k)$ on the interval $[\tau h, \tau' h]$ (for more details, see Section 3.5 in [144]).

2.5.2 Updating Formulas

The main question about the general quasi-Newton scheme is, of course, how to update Hessian approximations G_k (or their inverses H_k) at each iteration k. Needless to say, there are many ways to do this, each of which leads to a specific quasi-Newton method. Let us review the three most popular updating formulas, namely, SR1, DFP and BFGS.

In what follows, we consider the update at one particular iteration $k \geq 0$. To simplify our notation, we drop the index k everywhere and denote

$$\begin{aligned} x &:= x_k, & G &:= G_k, & H &:= H_k, \\ x_+ &:= x_{k+1}, & G_+ &:= G_{k+1}, & H_+ &:= H_{k+1}. \end{aligned}$$

Our goal is to describe how to update the self-adjoint positive definite linear operators

$$G \approx \nabla^2 f(x) \qquad \text{and} \qquad H \equiv G^{-1} \approx [\nabla^2 f(x)]^{-1}$$

into new self-adjoint positive definite linear operators

$$G_+ \approx \nabla^2 f(x_+) \qquad \text{and} \qquad H_+ \equiv G_+^{-1} \approx [\nabla^2 f(x_+)]^{-1}$$

by using the first-order information about f gathered at x and x_+:

$$\nabla f(x) \qquad \text{and} \qquad \nabla f(x_+).$$

Note that the point x_+ is already known at the moment when G_+ and H_+ are being computed.

Secant Equation

In classical quasi-Newton methods, the new Hessian approximation G_+ is required to satisfy the *secant equation*:

$$G_+ \delta = \gamma, \tag{2.5.5}$$

where

$$\delta := x_+ - x, \qquad \gamma := \nabla f(x_+) - \nabla f(x). \tag{2.5.6}$$

The motivation stems from the fact that, for a quadratic function f, the secant equation is satisfied by the exact Hessian $A := \nabla^2 f(x_+)$ ($= \nabla^2 f(x)$ for all $x \in \mathbb{E}$), i.e., $A\delta = \gamma$. For a general f, the secant equation is satisfied by the integral Hessian

$$J := \int_0^1 \nabla^2 f(x + t\delta) dt \tag{2.5.7}$$

which locally approximates $\nabla^2 f(x_+)$.

Note that the secant equation (2.5.5) alone is not enough to completely

specify G_+. Therefore, some other considerations are needed. One reasonable idea is to require that the difference between G_+ and G has *low rank*. In this case, one can cheaply compute H_+ using H without the need for explicitly inverting G_+.

SR1 Update

The simplest option is to require that G_+ differs from G by a rank-one self-adjoint linear operator and, at the same time, satisfies the secant equation (2.5.5). As it turns out, there exists only one formula, satisfying these requirements, namely, the *SR1 formula*:

$$G_+ = \mathrm{SR1}(G, \delta, \gamma) \coloneqq G - \frac{(G\delta - \gamma)(G\delta - \gamma)^*}{\langle G\delta - \gamma, \delta \rangle}, \tag{2.5.8}$$

which is defined whenever $\langle G\delta - \gamma, \delta \rangle \neq 0$. More precisely, we have the following result.

Lemma 2.5.1. *Suppose that G does not satisfy the secant equation (2.5.5), i.e., $G\delta \neq \gamma$. Then, among all self-adjoint rank-one corrections of G, there exists one, which satisfies the secant equation (2.5.5), iff $\langle G\delta - \gamma, \delta \rangle \neq 0$. When such a correction exists, it is unique and given by the SR1 formula (2.5.8).*

Proof. Consider an arbitrary self-adjoint rank-one correction of G:

$$G_+ \coloneqq G - \alpha s s^*, \tag{2.5.9}$$

where $s \in \mathbb{E}^*$ and $\alpha \in \mathbb{R}$. Suppose it satisfies the secant equation (2.5.5):

$$\gamma = G_+\delta = G\delta - \alpha \langle s, \delta \rangle s.$$

Then,

$$\alpha \langle s, \delta \rangle s = G\delta - \gamma. \tag{2.5.10}$$

Since $G\delta - \gamma \neq 0$, we have $\alpha \langle s, \delta \rangle \neq 0$. Thus, s is proportional to $G\delta - \gamma$:

$$s = t(G\delta - \gamma) \tag{2.5.11}$$

for some $t \in \mathbb{R}$. Substituting this representation into (2.5.10) and using the fact that $G\delta - \gamma \neq 0$, we obtain

$$\alpha t^2 \langle G\delta - \gamma, \delta \rangle = 1. \tag{2.5.12}$$

This is possible only if $\langle G\delta - \gamma, \delta \rangle \neq 0$, in which case, from (2.5.9), (2.5.11) and (2.5.12), we obtain

$$G - G_+ = \alpha s s^* = \alpha t^2 (G\delta - \gamma)(G\delta - \gamma)^* = \frac{(G\delta - \gamma)(G\delta - \gamma)^*}{\langle G\delta - \gamma, \delta \rangle}.$$

This is exactly the SR1 formula (2.5.8). □

Using the Sherman–Morrison identity, we can easily obtain the updating formula for the inverse Hessian approximation, which corresponds to the SR1 update from (2.5.8).

Lemma 2.5.2. *The inverse update, corresponding to* (2.5.8)*, is*

$$H_+ = \mathrm{SR1}^{-1}(H, \delta, \gamma) := H + \frac{(\delta - H\gamma)(\delta - H\gamma)^*}{\langle \gamma, \delta - H\gamma \rangle}, \tag{2.5.13}$$

which is defined whenever $\langle \gamma, \delta - H\gamma \rangle \neq 0$*.*

Proof. Indeed, by Proposition 2.1.1, applied to (2.5.8), we have

$$\begin{aligned} H_+ &= H + \frac{H(G\delta - \gamma)(G\delta - \gamma)^* H}{\langle G\delta - \gamma, \delta \rangle} \left[1 - \frac{\langle G\delta - \gamma, H(G\delta - \gamma) \rangle}{\langle G\delta - \gamma, \delta \rangle} \right]^{-1} \\ &= H + \frac{(\delta - H\gamma)(\delta - H\gamma)^*}{\langle G\delta - \gamma, H\gamma \rangle} = H + \frac{(\delta - H\gamma)(\delta - H\gamma)^*}{\langle \gamma, \delta - H\gamma \rangle}, \end{aligned}$$

where we have used the fact that $H \equiv G^{-1}$ and $H_+ \equiv G_+^{-1}$. □

The SR1 updating formula is quite efficient in practice. However, it is not particularly stable since the denominator in (2.5.8) and (2.5.13) may approach zero during the iterations of the method. Another drawback of the SR1 formula is that it does not preserve positive definiteness of Hessian approximations, which is highly desirable to guarantee that the direction (2.5.3), produced by the quasi-Newton method, can be used for decreasing the function value.

Least Change Principle

More stable updating formulas, which also preserve positive definiteness, can be obtained by considering *rank-two* corrections. In contrast to rank-one corrections, there exist a whole class of such formulas. Let us present one general approach which can be used for deriving them.

The approach, that we are going to present, is called the *least change principle*. The idea is to choose the new Hessian approximation G_+ in such a way that, on the one hand, it satisfies the secant equation (2.5.5), and, on the other hand, it is as close as possible to the current Hessian approximation G. This is indeed reasonable since, in this case, the second-order information, already accumulated in G, will not be completely destroyed, and the update will simply slightly correct it by adding some new second-order information, obtained from the current secant equation.

Denoting by $\beta(G, G_+)$ the "distance" between the current Hessian approximation G and some trial one G_+, we thus come to the following *Least Change Problem (LCP)*:

$$\min_{G_+ \in \operatorname{dom} d} \{\beta(G, G_+) : G_+\delta = \gamma\}, \tag{2.5.14}$$

where $\operatorname{dom} d$ is a certain feasible set in the space $\mathcal{S}(\mathbb{E}, \mathbb{E}^*)$, in which Hessian approximations are allowed to vary. Depending on the choice of the "distance" function β, we can obtain different specific formulas for the new Hessian approximation G_+ as the solutions of the LCP (2.5.14).

One way to define a meaningful and rather general notion of a "distance" is by using the *Bregman divergence*. Specifically, let us fix a certain function

$$d\colon \mathcal{S}(\mathbb{E}, \mathbb{E}^*) \to \mathbb{R} \cup \{+\infty\}$$

with open effective domain $\operatorname{dom} d$, on which d is differentiable and strictly convex. In what follows, we call such a function d a *prox function*. The *Bregman divergence*, generated by d, is the function $\beta\colon \operatorname{dom} d \times \operatorname{dom} d \to \mathbb{R}$, defined by

$$\beta(G, G_+) \coloneqq d(G_+) - d(G) - \langle \nabla d(G), G_+ - G \rangle. \tag{2.5.15}$$

Since d is strictly convex, the Bregman divergence $\beta(G, G_+)$ is nonnegative for any $G, G_+ \in \operatorname{dom} d$, and equals zero if and only if $G = G_+$. Thus, the value of $\beta(G, G_+)$ can be seen as a certain "distance" between $G, G_+ \in \operatorname{dom} d$. However, it is not a distance in the strict sense since the Bregman divergence, in general, is not symmetric.

Note that, by construction, $\beta(G, \cdot)$ is strictly convex for any $G \in \operatorname{dom} d$. Therefore, the LCP (2.5.14) is a convex optimization problem which admits at most one solution.

Let us present an optimality condition for the LCP (2.5.14).

Lemma 2.5.3. *An operator $G_+ \in \operatorname{dom} d$ is a solution of the LCP (2.5.14) iff there exists $\lambda \in \mathbb{E}$ such that*

$$\nabla d(G_+) = \nabla d(G) + \lambda\delta^* + \delta\lambda^*, \qquad G_+\delta = \gamma.$$

Proof. This is a simple corollary of the Lagrange multiplier rule. Indeed, let $L\colon \operatorname{dom} d \times \mathbb{E} \to \mathbb{R}$ be the Lagrange function

$$L(G_+, \lambda) := \beta(G, G_+) - 2\langle \lambda, G\delta - \gamma \rangle. \tag{2.5.16}$$

Then, $G_+ \in \operatorname{dom} d$ is a solution of LCP (2.5.14) iff $G_+\delta = \gamma$ and there exists $\lambda \in \mathbb{E}$ such that $\nabla_1 L(G_+, \lambda) = 0$, where $\nabla_1 L$ denotes the partial derivative of L w.r.t. its first argument. Recall that we work in the space $\mathcal{S}(\mathbb{E}, \mathbb{E}^*)$, therefore $\nabla_1 L(G_+, \lambda) \in [\mathcal{S}(\mathbb{E}, \mathbb{E}^*)]^* = \mathcal{S}(\mathbb{E}^*, \mathbb{E})$ (see (2.1.23)). Differentiating (2.5.16) and using (2.5.15), we obtain

$$\nabla_1 L(G_+, \lambda) = \nabla d(G_+) - \nabla d(G) - \lambda\delta^* - \delta\lambda^*$$

for any $G_+ \in \operatorname{dom} d$ and any $\lambda \in \mathbb{E}$. □

Now let us consider two specific examples of the prox function d, which lead to the most popular rank-two quasi-Newton updating formulas, namely, DFP and BFGS.

DFP Update

First, let us choose the *Euclidean prox function*

$$d(G) := \frac{1}{2}\|G\|^2_{\mathrm{F}(A)} \equiv \frac{1}{2}\langle A^{-1}GA^{-1}, G\rangle, \qquad \operatorname{dom} d \equiv \mathcal{S}(\mathbb{E}, \mathbb{E}^*), \tag{2.5.17}$$

where $\|G\|_{\mathrm{F}(A)}$ is the Frobenius norm of G w.r.t. a certain fixed *scaling operator* $A \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$. For this function, we have

$$\nabla d(G) = A^{-1}GA^{-1}, \tag{2.5.18}$$

and

$$\begin{aligned}\beta(G, G_+) &= \frac{1}{2}\|G_+\|^2_{\mathrm{F}(A)} - \frac{1}{2}\|G\|^2_{\mathrm{F}(A)} - \langle A^{-1}GA^{-1}, G_+ - G\rangle \\ &= \frac{1}{2}\|G_+\|^2_{\mathrm{F}(A)} - \langle A^{-1}GA^{-1}, G_+\rangle + \frac{1}{2}\|G\|^2_{\mathrm{F}(A)}\end{aligned}$$

$$= \frac{1}{2}\|G_+ - G\|^2_{\mathrm{F}(A)}.$$

Thus, the Bregman divergence is simply the squared Euclidean distance between G and G_+ (w.r.t. A).

Lemma 2.5.4. *The solution of the LCP* (2.5.14) *for the Euclidean prox function* (2.5.17) *is given by the following formula:*

$$G_+ = G - \frac{(G\delta - \gamma)\delta^* A + A\delta(G\delta - \gamma)^*}{\langle A\delta, \delta\rangle} + \frac{\langle G\delta - \gamma, \delta\rangle}{\langle A\delta, \delta\rangle^2} A\delta\delta^* A, \qquad (2.5.19)$$

assuming that $\delta \neq 0$.

Proof. According to Lemma 2.5.3 and (2.5.18), $G_+ \in \mathcal{S}(\mathbb{E}, \mathbb{E}^*)$ is a solution of LCP (2.5.14) iff there exists $\lambda \in \mathbb{E}$ such that

$$A^{-1}G_+A^{-1} = A^{-1}GA^{-1} + \lambda\delta^* + \delta\lambda^*, \qquad G_+\delta = \gamma,$$

or, equivalently,

$$G_+ = G + A\lambda\delta^* A + A\delta\lambda^* A, \qquad G_+\delta = \gamma. \qquad (2.5.20)$$

Substituting the formula for G_+ from the first equation into the second one, we obtain

$$\gamma = G\delta + \langle A\delta, \delta\rangle A\lambda + \langle A\lambda, \delta\rangle A\delta = G\delta + A_\delta\lambda, \qquad (2.5.21)$$

where $A_\delta \coloneqq \langle A\delta, \delta\rangle A + A\delta\delta^* A$. By the Sherman–Morrison formula (Proposition 2.1.1), we have

$$A_\delta^{-1} = \frac{A^{-1}}{\langle A\delta, \delta\rangle} - \frac{\delta\delta^*}{2\langle A\delta, \delta\rangle^2}.$$

Thus, equation (2.5.21) has a unique solution, namely,

$$\lambda = -A_\delta^{-1}(G\delta - \gamma) = \frac{\langle G\delta - \gamma, \delta\rangle\delta}{2\langle A\delta, \delta\rangle^2} - \frac{A^{-1}(G\delta - \gamma)}{\langle A\delta, \delta\rangle}. \qquad (2.5.22)$$

Substituting (2.5.22) into the first part of (2.5.20), we obtain (2.5.19). □

To make the updating formula from Lemma 2.5.4 *easily computable*, we need to choose A in such a way so that $A\delta$ is easily computable. At the same time, it is reasonable to make the resulting formula *affine-invariant*.

Arguably, the most natural way to ensure that both these requirements are satisfied is to demand that the scaling operator A satisfies the secant equation:

$$A\delta = \gamma. \tag{2.5.23}$$

For concreteness, we can assume that A is the integral Hessian from (2.5.7).

Combining Lemma 2.5.4 with our assumption (2.5.23), we come to the *DFP formula*:

$$\begin{aligned} G_+ &= \mathrm{DFP}(G, \delta, \gamma) \\ &\coloneqq G - \frac{(G\delta - \gamma)\gamma^* + \gamma(G\delta - \gamma)^*}{\langle \gamma, \delta \rangle} + \frac{\langle G\delta - \gamma, \delta \rangle}{\langle \gamma, \delta \rangle^2} \gamma\gamma^* \\ &= G - \frac{G\delta\gamma^* + \gamma\delta^* G}{\langle \gamma, \delta \rangle} + \Big(\frac{\langle G\delta, \delta \rangle}{\langle \gamma, \delta \rangle} + 1 \Big) \frac{\gamma\gamma^*}{\langle \gamma, \delta \rangle}, \end{aligned} \tag{2.5.24}$$

which is well-defined whenever $\langle \gamma, \delta \rangle > 0$.

Note that, in contrast to the SR1 formula (2.5.8), the denominator in the DFP formula (2.5.24) is always guaranteed to be positive whenever $\delta \neq 0$:

$$\delta \neq 0 \quad \Longrightarrow \quad \langle \gamma, \delta \rangle > 0.$$

This follows from the definitions of γ and δ in (2.5.6) and our assumption, made at the beginning of Section 2.5, that the Hessian of f is strictly positive definite. The case $\delta = 0$ is not especially interesting and corresponds to the situation when x_+ is an exact minimizer of f.

An important property of the DFP update is that it preserves positive definiteness, even though it was not explicitly required by our choice of the prox function (2.5.17). One simple way to see this is to rewrite the DFP update (2.5.24) in the following form:

$$\mathrm{DFP}(G, \delta, \gamma) = \Big(I_{\mathbb{E}^*} - \frac{\gamma\delta^*}{\langle \gamma, \delta \rangle} \Big) G \Big(I_{\mathbb{E}} - \frac{\delta\gamma^*}{\langle \gamma, \delta \rangle} \Big) + \frac{\gamma\gamma^*}{\langle \gamma, \delta \rangle}, \tag{2.5.25}$$

where $I_{\mathbb{E}}$ and $I_{\mathbb{E}^*}$ are the identity operators in $\mathbb{E}$ and $\mathbb{E}^*$, and $\langle \gamma, \delta \rangle > 0$.

To obtain the update for the inverse Hessian approximation, corresponding to the DFP formula, one can, in principle, apply the Sherman–Morrison formula twice to the representation (2.5.20) (where λ is given by (2.5.22) and A satisfies (2.5.23)). However, the corresponding computations are rather cumbersome. On the other hand, given a specific formula, it is quite easy to verify that it is indeed the correct inverse formula by doing a direct

multiplication.

Lemma 2.5.5. *The inverse update, corresponding to* (2.5.24)*, is*

$$H_+ = \mathrm{DFP}^{-1}(H, \delta, \gamma) := H - \frac{H\gamma\gamma^* H}{\langle \gamma, H\gamma \rangle} + \frac{\delta\delta^*}{\langle \gamma, \delta \rangle}, \tag{2.5.26}$$

provided that[15] $\langle \gamma, \delta \rangle > 0$.

Proof. Denote $G_+ := \mathrm{DFP}(G, \delta, \gamma)$ and $H_+ := \mathrm{DFP}^{-1}(H, \delta, \gamma)$. Let us prove that $G_+ H_+ = I_{\mathbb{E}^*}$ assuming that $H = G^{-1}$. From (2.5.26), it follows that $H_+ \gamma = \delta$. Combining this with (2.5.25), we obtain

$$\begin{aligned} G_+ H_+ &= \Big(I_{\mathbb{E}^*} - \frac{\gamma\delta^*}{\langle \gamma, \delta \rangle}\Big) G \Big(H_+ - \frac{\delta\delta^*}{\langle \gamma, \delta \rangle}\Big) + \frac{\gamma\delta^*}{\langle \gamma, \delta \rangle} \\ &= \Big(I_{\mathbb{E}^*} - \frac{\gamma\delta^*}{\langle \gamma, \delta \rangle}\Big) G \Big(H - \frac{H\gamma\gamma^* H}{\langle \gamma, H\gamma \rangle}\Big) + \frac{\gamma\delta^*}{\langle \gamma, \delta \rangle} = I_{\mathbb{E}^*}, \end{aligned}$$

where the second identity follows from (2.5.26). □

BFGS Update

Now consider the *log-det prox function*[16]:

$$d(G) := -\ln \det(A^{-1}, G), \qquad \operatorname{dom} d := \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*), \tag{2.5.27}$$

where $A \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ is a fixed scaling operator. The actual choice of A is not especially important since the resulting Bregman divergence turns out to be independent of A. Indeed, differentiating, we obtain

$$\nabla d(G) = -G^{-1}. \tag{2.5.28}$$

Therefore, in view of (2.5.15) and Propositions 2.1.4(iii) and 2.1.4(iv),

$$\begin{aligned} &\beta(G, G_+) \\ &\quad = -\ln \det(A^{-1}, G_+) + \ln \det(A^{-1}, G) - \langle -G^{-1}, G_+ - G \rangle \\ &\quad = \langle G^{-1}, G_+ - G \rangle - \ln \det(G^{-1}, G_+). \end{aligned} \tag{2.5.29}$$

Observe that, in contrast to the Euclidean prox function (2.5.17), which was defined on the entire space $\mathcal{S}(\mathbb{E}, \mathbb{E}^*)$, the log-det prox function is defined

[15] Recall that H is assumed to be positive definite. Hence, $\langle \gamma, H\gamma \rangle > 0$ whenever $\langle \gamma, \delta \rangle > 0$.

[16] Recall that $\det(\cdot, \cdot)$ is the determinant product defined in (2.1.31).

only on the cone $\mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ of positive definite linear operators. Thus, now the solution of the LCP (2.5.14) is explicitly required to be positive definite by our choice of the prox function.

Let us present the resulting updating formulas.

Lemma 2.5.6. *Suppose that $\langle \gamma, \delta \rangle > 0$. Then, the solution of LCP (2.5.14) with the log-det prox function (2.5.27) is given by the* BFGS formula[17]*:*

$$G_+ = \mathrm{BFGS}(G, \delta, \gamma) \coloneqq G - \frac{G\delta\delta^* G}{\langle G\delta, \delta \rangle} + \frac{\gamma\gamma^*}{\langle \gamma, \delta \rangle}. \tag{2.5.30}$$

The corresponding inverse update is given by

$$\begin{aligned} H_+ &= \mathrm{BFGS}^{-1}(H, \delta, \gamma) \\ &\coloneqq H - \frac{H\gamma\delta^* + \delta\gamma^* H}{\langle \gamma, \delta \rangle} + \left(\frac{\langle \gamma, H\gamma \rangle}{\langle \gamma, \delta \rangle} + 1 \right) \frac{\delta\delta^*}{\langle \gamma, \delta \rangle}. \end{aligned} \tag{2.5.31}$$

Proof. According to Lemma 2.5.3 and (2.5.28), $G_+ \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ is a solution of LCP (2.5.14) iff there exists $u \in \mathbb{E}$ such that[18]

$$-G_+^{-1} = -G^{-1} - u\delta^* - \delta u^*, \qquad G_+\delta = \gamma.$$

Rewriting these equations in terms of $H \equiv G^{-1}$ and $H_+ \equiv G_+^{-1}$, we obtain

$$H_+ = H + u\delta^* + \delta u^*, \qquad H_+\gamma = \delta.$$

Note that this system of linear equations has exactly the same structure as the one in (2.5.20). Substituting the formula for H_+ from the first equation into the second one and solving it, we find that

$$u = \frac{\langle \gamma, H\gamma - \delta \rangle \delta}{2\langle \gamma, \delta \rangle^2} - \frac{H\gamma - \delta}{\langle \gamma, \delta \rangle}.$$

Therefore,

$$H_+ = H - \frac{(H\gamma - \delta)\delta^* + \delta(H\gamma - \delta)^*}{\langle \gamma, \delta \rangle} + \frac{\langle \gamma, H\gamma - \delta \rangle}{\langle \gamma, \delta \rangle^2} \delta\delta^*,$$

and (2.5.31) follows after rearranging. To prove (2.5.30), one can apply a similar argument to that from the proof of Lemma 2.5.5. □

[17] Recall that G is assumed to be positive definite.
[18] In the notation of Lemma 2.5.3, $u = -\lambda$.

Comparing the BFGS and DFP updating formulas, we see that they are *dual* to each other: the direct BFGS formula (2.5.30) coincides with the inverse DFP formula (2.5.26) under the formal change of variables

$$G \leftrightarrow H, \qquad G_+ \leftrightarrow H_+, \qquad \delta \leftrightarrow \gamma, \tag{2.5.32}$$

and vice versa. In this sense, the SR1 formula is self-dual.

Directly Approximating Inverse Hessian

Let us briefly provide the duality relations, which we observed above, with one more interpretation.

Note that, instead of posing the LCP for the Hessian approximations, as we did in (2.5.14), we could, in fact, do the same directly for the *inverse* Hessian approximations:

$$\min_{H_+ \in \operatorname{dom} d_*} \{\beta_*(H, H_+) : H_+\gamma = \delta\}, \tag{2.5.33}$$

where $d_* \colon \mathcal{S}(\mathbb{E}^*, \mathbb{E}) \to \mathbb{R} \cup \{+\infty\}$ is a certain prox function, and β_* is the corresponding Bregman divergence:

$$\beta_*(H, H_+) \coloneqq d_*(H_+) - d_*(H) - \langle H_+ - H, \nabla d_*(H)\rangle. \tag{2.5.34}$$

As we see, LCP (2.5.33) is completely analogous to that from (2.5.14) under the formal change of variables (2.5.32) and the formal change of the prox function $d \leftrightarrow d_*$. Therefore, the following result should not be too surprising.

Lemma 2.5.7. *Suppose that $\langle \gamma, \delta \rangle > 0$. Then:*

(i) *The solution of LCP (2.5.33) for the Euclidean prox function*

$$d_*(H) \coloneqq \frac{1}{2}\|H\|^2_{\mathrm{F}(A)} \equiv \frac{1}{2}\langle H, AHA\rangle, \quad \operatorname{dom} d_* \coloneqq \mathcal{S}(\mathbb{E}^*, \mathbb{E}), \tag{2.5.35}$$

where $A \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ is a fixed scaling operator, satisfying secant equation (2.5.23), is given by the BFGS formula (2.5.31).

(ii) *The solution of LCP (2.5.33) for the log-det prox function*

$$d_*(H) \coloneqq -\ln\det(H, A) - n, \qquad \operatorname{dom} d_* \coloneqq \mathcal{S}_{++}(\mathbb{E}^*, \mathbb{E}), \tag{2.5.36}$$

where $A \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ is an arbitrary scaling operator, is given by the DFP update (2.5.26).

Note that the constant term "$-n$" in (2.5.36) does not play any significant role since it is cancelled inside the Bregman divergence (2.5.34). However, by keeping it, we obtain an interesting relation between the prox functions d_* from (2.5.35) and (2.5.36) and the corresponding prox functions d from (2.5.17) and (2.5.27), considered earlier. Specifically, they turn out to be duals: for any $H \in \mathcal{S}(\mathbb{E}^*, \mathbb{E})$, we have

$$d_*(H) = \sup_{G \in \operatorname{dom} d} \{\langle -H, G\rangle - d(G)\}.$$

Thus, d_* is the standard Fenchel conjugate function of d up to a minor change in the sign in front of the argument. Note also that the Euclidean norm, defined in (2.5.35), is actually *conjugate* to the one in (2.5.17).

Broyden Class

The SR1, DFP and BFGS updates, considered earlier, are all members of a more general one-parameter family of updating formulas. This family is called the *Broyden class*, and can be defined as the linear combination of the BFGS and DFP updates:

$$\operatorname{Broyd}_\varphi(G, \delta, \gamma) \coloneqq (1-\varphi)\operatorname{BFGS}(G, \delta, \gamma) + \varphi \operatorname{DFP}(G, \delta, \gamma), \qquad (2.5.37)$$

where $\varphi \in \mathbb{R}$ is a parameter.

Obviously, the BFGS and DFP updates correspond to $\varphi = 0$ and $\varphi = 1$, respectively. The SR1 update corresponds to

$$\varphi^{\mathrm{SR1}} \coloneqq -\frac{\langle \gamma, \delta\rangle}{\langle G\delta - \gamma, \delta\rangle},$$

provided that $\langle G\delta - \gamma, \delta\rangle \neq 0$. Indeed, denoting $G_+^{\mathrm{SR1}} \coloneqq \operatorname{Broyd}_{\varphi^{\mathrm{SR1}}}(G, \delta, \gamma)$ and using (2.5.30) and (2.5.24), we obtain

$$\begin{aligned} G_+^{\mathrm{SR1}} &= \frac{\langle G\delta, \delta\rangle}{\langle G\delta - \gamma, \delta\rangle}\left[G - \frac{G\delta\delta^* G}{\langle G\delta, \delta\rangle} + \frac{\gamma\gamma^*}{\langle \gamma, \delta\rangle}\right] \\ &\quad - \frac{\langle \gamma, \delta\rangle}{\langle G\delta - \gamma, \delta\rangle}\left[G - \frac{G\delta\gamma^* + \gamma\delta^* G}{\langle \gamma, \delta\rangle} + \Big(\frac{\langle G\delta, \delta\rangle}{\langle \gamma, \delta\rangle} + 1\Big)\frac{\gamma\gamma^*}{\langle \gamma, \delta\rangle}\right] \\ &= G - \frac{G\delta\delta^* G - (G\delta\gamma^* + \gamma\delta^* G) + \gamma\gamma^*}{\langle G\delta - \gamma, \delta\rangle} \end{aligned}$$

$$= G - \frac{(G\delta - \gamma)(G\delta - \gamma)^*}{\langle G\delta - \gamma, \delta \rangle},$$

which is exactly the SR1 formula (2.5.8).

Note that any member of the Broyden class (2.5.37) satisfies the secant equation (2.5.5) since so do the BFGS and DFP updates. However, in general, the Broyden update does not preserve positive definiteness.

An important subclass of the Broyden class (2.5.37) corresponds to the values of $\varphi \in [0, 1]$. It consists of all convex combinations of the BFGS and DFP updates, and is called the *convex Broyden class*. Every member of this class satisfies the secant equation (2.5.37), preserves positive definiteness and has other interesting properties which we will study further in Section 3.1. Note that the SR1 update, in general, does not belong to the convex Broyden class.

2.5.3 Convergence Results

Let us now state the classical convergence results about the BFGS and DFP methods, i.e., the instances of the general quasi-Newton scheme from Algorithm 2.5.1, resulting by using the BFGS and DFP formulas, respectively, for updating the inverse Hessian approximation at each iteration. Recall that our problem under consideration is (2.5.1).

Algorithm 2.5.3: BFGS/DFP Method

Initialization: Choose $x_0 \in \mathbb{E}$ and $H_0 \in \mathcal{S}_{++}(\mathbb{E}^*, \mathbb{E})$.

Iteration $k \geq 0$**:**

1. Choose a step size $h_k \geq 0$.
2. Set $x_{k+1} \coloneqq x_k - h_k H_k \nabla f(x_k)$.
3. Compute $\delta_k \coloneqq x_{k+1} - x_k$ and $\gamma_k \coloneqq \nabla f(x_{k+1}) - \nabla f(x_k)$.
4. Update inverse Hessian approximation:

   (BFGS Method) $H_{k+1} \coloneqq \mathrm{BFGS}^{-1}(H_k, \delta_k, \gamma_k)$.

   (DFP Method) $H_{k+1} \coloneqq \mathrm{DFP}^{-1}(H_k, \delta_k, \gamma_k)$.

The first result is about *local convergence* and is due to Broyden, Dennis and Moré [20].

Theorem 2.5.8. *Suppose*[19] *that the function f is strongly convex and has Lipschitz continuous Hessian. Consider either the BFGS or DFP Method from Algorithm 2.5.3 with unit step sizes:*

$$h_k \equiv 1, \qquad k \geq 0.$$

Then, for every $\rho \in (0,1)$, there exist $\delta_1, \delta_2 > 0$ such that, for any initial point x_0 and any initial inverse Hessian approximation H_0, satisfying

$$\|x_0 - x^*\| \leq \delta_1, \qquad \|H_0 - [\nabla^2 f(x^*)]^{-1}\| \leq \delta_2,$$

where x^* is the solution of (2.5.1), we have, for all $k \geq 0$,

$$\|x_{k+1} - x^*\| \leq \rho \|x_k - x^*\|.$$

Moreover, the rate of convergence is asymptotically superlinear:

$$\lim_{k \to \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0.$$

The assumptions in Theorem 2.5.8 are quire similar to those which are used in the classical analysis of Newton's Method (see Section 2.4.1). The main difference is that now one requires that both the initial point x_0 and the initial inverse Hessian approximation H_0 are sufficiently good. The latter assumption was redundant in Newton's Method since, in that algorithm, the closeness of $H_0 = [\nabla^2 f(x_0)]^{-1}$ to $[\nabla^2 f(x^*)]^{-1}$ automatically followed from the closeness of x_0 to x^*.

For the BFGS Method, we also have the following *global* convergence result. The version with the Wolfe conditions was first proved by Powell [150], while the version with the backtracking line search is due to Byrd and Nocedal [24].

Theorem 2.5.9. *Suppose that the function f is strongly convex and has Lipschitz continuous gradient and Hessian. Consider the BFGS Method from Algorithm 2.5.3, which uses any of the following line search strategies for choosing the step size h_k at each iteration $k \geq 0$:*

1. *either the line search, satisfying the Wolfe conditions (2.5.4),*

[19] In fact, it suffices to assume that $\nabla^2 f(x^*)$ is nonsingular and the Hessian of f is Lipschitz continuous only w.r.t. the fixed point x^*, i.e., $\|\nabla^2 f(x) - \nabla^2 f(x^*)\| \leq L_2 \|x - x^*\|$ for some $L_2 \geq 0$ and all x from a certain neighborhood of x^*. However, for the sake of simplicity, we slightly relax these assumptions.

2. or the backtracking line search from Algorithm 2.5.2.

Then, for any initial point x_0 and any initial inverse Hessian approximation H_0, the sequence x_k converges to the solution x^* of (2.5.1):

$$\lim_{k \to \infty} x_k = x^*.$$

Moreover, the rate of convergence is asymptotically superlinear:

$$\lim_{k \to \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0.$$

Interestingly, there is no counterpart of Theorem 2.5.9 for the DFP Method, and it is currently unknown whether the DFP Method with an inexact line search is globally convergent at all. Nevertheless, there exists an extension of Theorem 2.5.9 to the entire convex Broyden class except the DFP Method [25].

Looking at Theorems 2.5.8 and 2.5.9, we see that they are only qualitative in that they do not provide us with any *explicit estimates* of the *rate* of convergence. As a consequence, we cannot really use these theorems for deriving the iteration complexity bounds for obtaining an ε-approximate solution to problem (2.5.1) by the BFGS and DFP methods, similar to those bounds we had for Newton's Method in Section 2.4. This is exactly the reason why we criticized these classical results in Section 1.1. We will return to this issue and address it in more detail in Chapter 3.

Note that the requirements in Theorem 2.5.8 are quite strong: both the initial point and the initial Hessian approximation must be sufficiently good. At the same time, Theorem 2.5.9 works for any initial point and any initial Hessian approximation. However, in contrast to Theorem 2.5.8, it is assumed that the method uses line search. In Chapter 3, we will present some intermediate result, namely, a version of Theorem 2.5.8 with a much weaker assumption on the initial Hessian approximation.

Finally, let us mention that, for the SR1 Method, no convergence results, similar to those from Theorems 2.5.8 and 2.5.9, have been established. However, in Chapter 4, we will see that for a certain version of this method, which uses a special correction strategy for keeping the Hessian approximation above the actual Hessian, it is still possible to prove local superlinear convergence.

2.6 Subgradient Method

Now we switch our attention to a different problem formulation:

$$\min_{x \in Q} f(x), \tag{2.6.1}$$

where $Q \subseteq \mathbb{E}$ is a closed convex set, and $f\colon \mathbb{E} \to \mathbb{R}$ is a general *nonsmooth* convex function. We assume that problem (2.6.1) has a solution x^* and denote by f^* the corresponding optimal value.

Our main assumption about the objective function in problem (2.6.1) is that it is Lipschitz with some constant $M > 0$, i.e., for all $x, y \in \mathbb{E}$,

$$|f(x) - f(y)| \leq M\|x - y\|, \tag{2.6.2}$$

where $\|\cdot\|$ is a Euclidean norm, generated by some $B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$:

$$\|x\| \coloneqq \|x\|_B \coloneqq \langle Bx, x\rangle^{1/2}, \qquad x \in \mathbb{E}.$$

Let us consider the Subgradient Method for solving (2.6.1). It uses the following projection operator onto the set Q:

$$\pi_Q(x) \coloneqq \operatorname*{argmin}_{y \in Q} \|x - y\|, \qquad x \in \mathbb{E}.$$

Note that $\pi_Q(x)$ is well-defined and unique for any $x \in \mathbb{E}$ since the set Q is assumed to be closed and convex.

Algorithm 2.6.1: Subgradient Method
Input: Initial point $x_0 \in Q$.
Iteration $k \geq 0$**:** 1. Compute an arbitrary subgradient $g_k \in \partial f(x_k)$. 2. Choose a step size $h_k > 0$. 3. Compute $x_{k+1} \coloneqq \pi_Q(x_k - h_k B^{-1} g_k / \|g_k\|_*)$.

We assume that the set Q and the operator B, defining the norm $\|\cdot\|$, are sufficiently simple so that both the inverse operator B^{-1} and the projection $\pi_Q(x)$ can be efficiently computed for any $x \in \mathbb{E}$. Further, without loss of generality, we assume that $g_k \neq 0$ for all $k \geq 0$ in Algorithm 2.6.1. The "unlikely" case when $g_k = 0$ for some $k \geq 0$ corresponds to the situation

when Algorithm 2.6.1 has "accidentally" found a global minimizer x_k of the function f.

The approximate solution to problem (2.6.1) generated by the Subgradient Method after $k \geq 1$ iterations is defined as the search point with the best function value found so far[20]:

$$x_k^* := \operatorname{argmin}\{f(x) : x \in \{x_0, \ldots, x_{k-1}\}\} \ (\in Q). \tag{2.6.3}$$

Let us present efficiency estimates for the Subgradient Method. We start with a general bound, which is valid for an arbitrary choice of step sizes.

Lemma 2.6.1. *Suppose*[21] *that* $\|x_0 - x^*\| \leq R$. *Then, for all* $k \geq 1$,

$$f(x_k^*) - f^* \leq M \frac{R^2 + \sum_{i=0}^{k-1} h_i^2}{2 \sum_{i=0}^{k-1} h_i}. \tag{2.6.4}$$

Proof. Let $0 \leq i < k$ be arbitrary integers. Using the fact that the projection is a contraction and denoting $\tilde{x}_{i+1} := x_i - h_i B^{-1} g_i / \|g_i\|_*$, we obtain

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &\leq \|\tilde{x}_{k+1} - x^*\|^2 \\ &= \|x_i - x^*\|^2 - 2h_i \|g_i\|_*^{-1} \langle g_i, x_i - x^* \rangle + h_i^2. \end{aligned}$$

Since $g_i \in \partial f(x_i)$ and in view of Proposition 2.2.12, we have

$$f(x_i) - f^* \leq \langle g_i, x_i - x^* \rangle \qquad \text{and} \qquad \|g_i\|_* \leq M.$$

Thus,

$$\|x_{k+1} - x^*\|^2 \leq \|x_i - x^*\|^2 - 2h_i M^{-1} [f(x_i) - f^*] + h_i^2. \tag{2.6.5}$$

Summing up these inequalities for all i from 0 to $k-1$, we obtain

$$2M^{-1} \sum_{i=0}^{k-1} h_i [f(x_i) - f^*] \leq \|x_0 - x^*\|^2 + \sum_{i=0}^{k-1} h_i^2 \leq R^2 + \sum_{i=0}^{k-1} h_i^2.$$

It remains to use the fact that $f(x_k^*) \leq f(x_i)$ for all $0 \leq i \leq k-1$ in view of (2.6.3). □

[20] Any possible ties in this definition are assumed to be resolved arbitrarily.

[21] Here x^* can be an arbitrary solution to (2.6.1). In other words, in the case when the solution set X^* of (2.6.1) is not a singleton, instead of $\|x_0 - x^*\| \leq R$, it suffices to assume that $\inf_{x^* \in X^*} \|x_0 - x^*\| \leq R$. The same remark applies throughout this section.

Let us now consider several standard strategies, which can be used for selecting the step sizes in Algorithm 2.6.1. The simplest one is the *constant step* strategy, which comes from minimizing the right-hand side in (2.6.4) w.r.t. $h_0, \dots, h_{k-1}$ for any fixed value of $k \geq 1$.

Theorem 2.6.2. *Suppose that $\|x_0 - x^*\| \leq R$ for some $R > 0$. Consider Algorithm 2.6.1 with the constant step sizes:*

$$h_k := h := \frac{R}{\sqrt{K}}, \qquad 0 \leq k \leq K-1,$$

where $K \geq 1$ is some predefined number of iterations, which the method is going to perform. Then,

$$f(x_K^*) - f^* \leq \frac{MR}{\sqrt{K}}.$$

Proof. This is a simple consequence of Lemma 2.6.1. □

In order to avoid doing a predefined number of iterations, one can choose *time-varying* step sizes. Then, the resulting convergence rate estimate will be slightly worse, but only by a logarithmic factor. To show this, let us first prove an auxiliary result.

Lemma 2.6.3. *Let $0 \leq k_0 < k$ be integer. Then,*

$$\sum_{i=k_0+1}^{k} \frac{1}{\sqrt{i}} \geq \frac{k-k_0}{\sqrt{k}}, \qquad \sum_{i=k_0+1}^{k} \frac{1}{i} \leq \ln \frac{k}{k_0} \tag{2.6.6}$$

(assuming $k_0 > 0$ for the second inequality). Consequently,

$$\sum_{i=1}^{k} \frac{1}{\sqrt{i}} \geq \sqrt{k}, \qquad \sum_{i=1}^{k} \frac{1}{i} \leq \ln k + 1. \tag{2.6.7}$$

Proof. The first inequality in (2.6.6) easily follows from the fact that $i \leq k$ for any $k_0 + 1 \leq i \leq k$. The second inequality is a consequence of the standard integral bound:

$$\sum_{i=k_0+1}^{k} \frac{1}{i} \leq \sum_{i=k_0+1}^{k} \int_{i-1}^{i} \frac{d\tau}{\tau} = \int_{k_0}^{k} \frac{d\tau}{\tau} = \ln \frac{k}{k_0}.$$

Substituting $k_0 = 0$ into the first inequality in (2.6.6) gives us the first

inequality in (2.6.7). To prove the second inequality, substitute $k_0 = 1$ into the second inequality in (2.6.6) and add 1 to both sides. □

Equipped with Lemma 2.6.3, we are now ready to present the efficiency estimate for Algorithm 2.6.1 with time-varying step sizes.

Theorem 2.6.4. *Suppose that $\|x_0 - x^*\| \leq R$ for some $R > 0$. Consider Algorithm 2.6.1 with step sizes*

$$h_k := \frac{R}{\sqrt{k+1}}, \qquad k \geq 0. \tag{2.6.8}$$

Then, for all $k \geq 1$, we have

$$f(x_k^*) - f^* \leq \frac{2 + \ln k}{2\sqrt{k}} MR.$$

Proof. Indeed, according to (2.6.8) and Lemma 2.6.3, for any $k \geq 1$,

$$\sum_{i=0}^{k-1} h_i^2 = R^2 \sum_{i=1}^{k} \frac{1}{i} \leq R^2(1 + \ln k),$$
$$\sum_{i=0}^{k-1} h_i = R \sum_{i=1}^{k} \frac{1}{\sqrt{i}} \geq R\sqrt{k}.$$

It remains to apply Lemma 2.6.1. □

In the important special case, when the feasible set Q in problem (2.6.1) is *bounded*, it becomes possible to get rid of the additional logarithmic factor altogether at the cost of replacing the constant R with a certain bound D on the "radius" of the set Q.

Theorem 2.6.5. *Suppose that Q is bounded: for some $D > 0$, we have*

$$\|x - x^*\| \leq D, \qquad \forall x \in Q. \tag{2.6.9}$$

Consider Algorithm 2.6.1 with step sizes

$$h_k := \frac{D}{\sqrt{k+1}}, \qquad k \geq 0. \tag{2.6.10}$$

Then, for all $k \geq 1$, we have

$$f(x_k^*) - f^* \leq (1 + \ln 3) \frac{MD}{\sqrt{k}}. \tag{2.6.11}$$

Proof. For any $k \geq 1$, in view of (2.6.2), (2.6.3) and (2.6.9), we have

$$f(x_k^*) - f^* \leq M\|x_k^* - x^*\| \leq MD.$$

Therefore, it suffices to prove (2.6.10) only for $k \geq 3$ (say).

Let $0 < k_0 < k$ be arbitrary integers with $k \geq 3$. Repeating the proof of Lemma 2.6.1, but now summing up (2.6.5) from k_0 (instead of 0) to $k-1$, we obtain

$$2M^{-1} \sum_{i=k_0}^{k-1} h_i [f(x_i) - f^*] \leq \|x_{k_0} - x^*\|^2 + \sum_{i=k_0}^{k-1} h_i^2 \leq D^2 + \sum_{i=k_0}^{k-1} h_i^2,$$

where the second inequality follows from (2.6.9) and the fact that $x_{k_0} \in Q$. Combining this result with (2.6.3), we get

$$\begin{aligned} f(x_k^*) - f^* &= \min_{0 \leq i \leq k-1} f(x_i) - f^* \\ &\leq \min_{k_0 \leq i \leq k-1} f(x_i) - f^* \leq M \frac{D^2 + \sum_{i=k_0}^{k-1} h_i^2}{2 \sum_{i=k_0}^{k-1} h_i}. \end{aligned} \tag{2.6.12}$$

Further, according to (2.6.10) and Lemma 2.6.3, we have

$$\begin{aligned} \sum_{i=k_0}^{k-1} h_i &= D \sum_{i=k_0+1}^{k} \frac{1}{\sqrt{i}} \geq \frac{k-k_0}{\sqrt{k}} D, \\ \sum_{i=k_0}^{k-1} h_i^2 &= D^2 \sum_{i=k_0+1}^{k} \frac{1}{i} \leq D^2 \ln \frac{k}{k_0}. \end{aligned}$$

Let us choose $k_0 := \lfloor k/2 \rfloor$. Then, $(k-1)/2 \leq k_0 \leq k/2$, and hence

$$\frac{k-k_0}{\sqrt{k}} \geq \frac{1}{2}\sqrt{k}, \qquad \frac{k}{k_0} \leq 2\frac{k}{k-1} \leq 3$$

since $k \geq 3$. Thus,

$$\sum_{i=k_0}^{k-1} h_i \geq \frac{1}{2} D\sqrt{k}, \qquad \sum_{i=k_0}^{k-1} h_i^2 \leq D^2 \ln 3.$$

Substituting these inequalities into (2.6.12), we obtain (2.6.11). □

According to Theorem 2.6.5, to find an ε-approximate solution (in terms

of function value) to problem (2.6.1), the Subgradient Method requires at most the following number of iterations:

$$O\Big(\frac{M^2 D^2}{\varepsilon^2}\Big), \tag{2.6.13}$$

where M is the Lipschitz constant of the objective function and D is an upper bound on the diameter of the feasible set Q.

2.7 Ellipsoid Method

Let us now review the Ellipsoid Method. We consider the same problem as before, namely,

$$\min_{x \in Q} f(x), \tag{2.7.1}$$

where $Q \subseteq \mathbb{E}$ is a closed convex set, and $f\colon \mathbb{E} \to \mathbb{R}$ is a general convex function. However, now we additionally assume that the feasible set Q is *bounded* and has *nonempty interior*. Thus, Q is a *solid* (compact convex set with nonempty interior). Note that, under our assumptions, the solution set in problem (2.7.1) is nonempty. We denote the corresponding optimal value by f^*.

Throughout this section, all efficiency estimates will be presented, among others, in terms of the following two parameters of problem (2.7.1):

- *Dimensionality* of the space:

$$n := \dim \mathbb{E}. \tag{2.7.2}$$

- *Variation* of the objective function on the feasible set:

$$V := \max_{x \in Q} f(x) - f^*. \tag{2.7.3}$$

The function f in problem (2.7.1), may, in general, be nonsmooth. We assume that it is represented by the standard *First-Order Oracle*: given any point $x \in \mathbb{E}$, it returns an arbitrary subgradient $f'(x)$ of f at x.

In contrast to the Subgradient Method from Section 2.6, in which all the iterates automatically belong to the feasible set Q, thanks to the projection, the Ellipsoid Method may sometimes produce points which lie outside Q. To handle such infeasible points, it uses a special *Separation Oracle* for the

set Q: given any point $x \in \mathbb{E}$, this oracle can check whether $x \in \operatorname{int} Q$, and if not, it reports a vector $g_Q(x) \in \mathbb{E}^* \setminus \{0\}$ which *separates* x from Q:

$$\langle g_Q(x), x - y \rangle \geq 0, \qquad \forall y \in Q. \tag{2.7.4}$$

For example, in the case when the set Q is specified by a certain general convex function $g\colon \mathbb{E} \to \mathbb{R}$, i.e., $Q \coloneqq \{x \in \mathbb{E} : g(x) \leq 0\}$, it is not difficult to see that, for any $x \in \mathbb{E}$, a separator is given by $g_Q(x) = g'(x)$, where $g'(x)$ is an arbitrary subgradient of g at x.

Thus, we have two oracles representing problem (2.7.1). For the sake of convenience, let us unite them into one: for any $x \in \mathbb{E}$, define

$$\mathcal{G}(x) \coloneqq \begin{cases} f'(x), & \text{if } x \in \operatorname{int} Q, \\ g_Q(x), & \text{otherwise.} \end{cases} \tag{2.7.5}$$

To avoid considering certain degenerate cases all the time, from now on, we will assume that the oracle (2.7.5) never returns zero. This is indeed the case for any point $x \notin \operatorname{int} Q$ by the definition of the Separation Oracle. Should it happen that $\mathcal{G}(x) = 0$ for some $x \in \operatorname{int} Q$, we can always stop the method and return x as the exact solution of problem (2.7.1).

The Ellipsoid Method is a particular instance of a more general family of algorithms, known as *cutting plane methods*. Let us briefly review the general scheme of these algorithms before presenting the Ellipsoid Method itself. For more details, we refer the reader to [122].

2.7.1 General Cutting Plane Scheme

The general cutting plane scheme for solving problem (2.7.1), equipped with oracle (2.7.5), is based on the idea of *localization*. Specifically, let x^* be a solution to problem (2.7.1). Then, for any $x \in \mathbb{E}$, we have

$$\langle \mathcal{G}(x), x - x^* \rangle \geq 0. \tag{2.7.6}$$

Indeed, if $x \in \operatorname{int} Q$, then $\mathcal{G}(x) = f'(x)$, and hence

$$\langle \mathcal{G}(x), x - x^* \rangle = \langle f'(x), x - x^* \rangle \geq f(x) - f^* \geq 0$$

since $f'(x) \in \partial f(x)$ and f^* is the minimal value of f on Q. If $x \notin \operatorname{int} Q$, then $\mathcal{G}(x) = g_Q(x)$ is a separator of x from Q, and (2.7.6) immediately follows from (2.7.4) since $x^* \in Q$.

Inequality (2.7.6) has a simple geometric meaning. It tells us that, for any point $x \in \mathbb{E}$, the oracle output $\mathcal{G}(x)$ provides us with a hyperplane, which divides the space $\mathbb{E}$ into two parts, only one of which contains the solution set of problem (2.7.1). This leads us to the following natural idea. Suppose that we already have a certain *localizer* Ω— a solid containing the solution set. Let us choose somehow a point $\bar{x} \in \Omega$ and then "cut" our localizer Ω with the hyperplane, passing through $\bar{x}$ with the normal vector $\mathcal{G}(\bar{x})$. As we already know, only one "half" of Ω will be containing the solution set. Therefore, it makes sense to take this "half" as the new localizer and repeat the procedure.

Formalizing the above considerations, we arrive at the following algorithmic scheme.

Algorithm 2.7.1: General Cutting Plane Scheme

Initialization: Choose a solid $\Omega_0 \supseteq Q$.

Iteration $k \geq 0$**:**

1. Choose a point $x_k \in \Omega_k$.
2. Query the oracle to obtain $g_k := \mathcal{G}(x_k)$.
3. Choose a solid $\Omega_{k+1} \supseteq \hat{\Omega}_{k+1} := \{x \in \Omega_k : \langle g_k, x_k - x \rangle \geq 0\}$.

By definition, the approximate solution to problem (2.7.1), generated by Algorithm 2.7.1 after $k \geq 1$ iterations, is the best among all feasible search points produced so far:

$$x_k^* := \operatorname{argmin}\bigl\{f(x) : x \in \{x_0, \ldots, x_{k-1}\} \cap \operatorname{int} Q\bigr\}. \tag{2.7.7}$$

If all points $x_0, \ldots, x_{k-1}$ are infeasible ($\notin \operatorname{int} Q$), we leave x_k^* undefined.

In the "pure" cutting plane scheme, the initial localizer is $\Omega_0 = Q$, and the new localizer Ω_{k+1}, at each iteration $k \geq 0$, is chosen to be exactly $\hat{\Omega}_{k+1}$ (the "half" of the current localizer Ω_k). However, in general, to keep the iteration cost at a reasonable level, it makes sense to keep the localizers Ω_k in some simple form, e.g., ellipsoids.

Algorithm 2.7.1 is a general scheme since it does not describe how exactly the points x_k and the localizers Ω_k are chosen at each iteration $k \geq 0$. By specifying these rules, we obtain a particular instance of the cutting plane scheme. In general, the goal is to choose x_k and Ω_k in such a way so that the "size" of the localizers Ω_k goes to zero sufficiently fast.

Of course, there exist many ways how one can measure the "size" of localizers. However, for any particular "size" we decide to work with, we need to be able to conclude that, whenever the "size" of a localizer is sufficiently small, the approximate solution (2.7.7), produced by Algorithm 2.7.1, is well-defined and is nearly optimal for problem (2.7.1). One rather general family of "sizes" which satisfies this requirement is as follows.

Definition 2.7.1. Let size: $\mathcal{Q} \to (0, +\infty)$ be a strictly positive function, defined on the collection $\mathcal{Q}$ of all solids in $\mathbb{E}$. Then, size is called a *size function* if it satisfies the following two requirements:

(i) (Monotonicity) For any $\Omega_1, \Omega_2 \in \mathcal{Q}$, such that $\Omega_1 \subseteq \Omega_2$, we have

$$\operatorname{size}(\Omega_1) \leq \operatorname{size}(\Omega_2).$$

(ii) (Homogeneity w.r.t. homotheties) For any $\Omega \in \mathcal{Q}$, $x \in \Omega$, $\alpha \in (0, 1)$,

$$\operatorname{size}((1-\alpha)x + \alpha\Omega) = \alpha \operatorname{size}(\Omega).$$

The simplest example of a size function is the standard *diameter*:

$$\operatorname{diam}\Omega := \max_{x,y\in\Omega} \|x - y\|,$$

where $\|\cdot\|$ is an arbitrary norm in $\mathbb{E}$. Another important example is the so-called *average radius*[22]:

$$\operatorname{avrad}\Omega := [\operatorname{vol}(\Omega/B_0)]^{1/n}, \tag{2.7.8}$$

where B_0 an arbitrary solid (e.g., the unit ball/cube) in the space $\mathbb{E}$. The monotonicity of the average radius follows from Proposition 2.1.7(ii), and the homogeneity—from Propositions 2.1.7(i) and 2.1.7(iv).

We are ready to present the main result about the general cutting plane scheme.

Theorem 2.7.2. *Consider some iteration $k \geq 1$ of Algorithm 2.7.1. Suppose that, for a certain size function, we have*

$$\delta_k := \frac{\operatorname{size}\Omega_k}{\operatorname{size} Q} < 1.$$

[22] Recall that $\operatorname{vol}(\cdot/\cdot)$ is the relative volume defined in (2.1.32).

Then, the approximate solution (2.7.7) *is well-defined and*

$$f(x_k^*) - f^* \leq \delta_k V. \tag{2.7.9}$$

Proof. Let $\delta \in (\delta_k, 1)$ and let x^* be a solution to (2.7.1). Consider the set

$$Q^\delta := (1-\delta)x^* + \delta Q.$$

By the homogeneity and strict positivity of the size function, we have

$$\operatorname{size}(Q^\delta) = \delta \operatorname{size} Q > \delta_k \operatorname{size} Q = \operatorname{size} \Omega_k.$$

Therefore, by the monotonicity, Q^δ cannot be a subset of Ω_k:

$$Q^\delta \not\subseteq \Omega_k.$$

In other words, there exists $x \in Q$ such that

$$z := (1-\delta)x^* + \delta x \notin \Omega_k. \tag{2.7.10}$$

Note that $z \in Q$ since Q is a convex set and $x, x^* \in Q$. Hence,

$$\langle g_i, x_i - z \rangle < 0 \tag{2.7.11}$$

for some $0 \leq i \leq k-1$ (otherwise, a simple inductive argument shows that $z \in \Omega_k$ which contradicts (2.7.10)). Clearly, for this index i, we must have $x_i \in \operatorname{int} Q$ (otherwise, (2.7.10) contradicts the separation property (2.7.4) since $z \in Q$). Thus, the approximate solution (2.7.7) is well-defined and $g_i = f'(x_i) \in \partial f(x_i)$. Consequently, from (2.7.11) and the definition of the subgradient, we obtain

$$f(x_i) < f(x_i) + \langle f'(x_i), z - x_i \rangle \leq f(z).$$

Thus, by (2.7.10), convexity of f and (2.7.3) (with the fact that $x \in Q$),

$$f(x_i) - f^* \leq f\big((1-\delta)x^* + \delta x\big) - f^* \leq \delta[f(x) - f^*] \leq \delta V.$$

Since this inequality is valid for some $0 \leq i \leq k-1$, such that $x_i \in \operatorname{int} Q$, we have, in view of (2.7.7),

$$f(x_k^*) - f^* \leq \delta V.$$

Taking now the limit in this inequality as $\delta \to \delta_k$, we obtain (2.7.9). □

2.7.2 Ellipsoid Method

The Ellipsoid Method is a particular implementation of the general cutting plane scheme, in which the localizers are *ellipsoids*, i.e., sets of the form

$$\mathcal{E}(\bar{x}, G) \coloneqq \{x \in \mathbb{E} : \|x - \bar{x}\|_G \leq 1\}, \tag{2.7.12}$$

where $\bar{x} \in \mathbb{E}$ and $G \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ are certain parameters. It is based on the following key result which describes how to implement Step 3 in Algorithm 2.7.1 efficiently.

Lemma 2.7.3. *Let $E \coloneqq \mathcal{E}(\bar{x}, G)$ be an ellipsoid, where $\bar{x} \in \mathbb{E}$ and $G \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$, and let $g \in \mathbb{E}^* \setminus \{0\}$. Then, the "half-ellipsoid"*

$$\bar{E} \coloneqq \{x \in E : \langle g, \bar{x} - x\rangle \geq 0\},$$

resulting by cutting E with a hyperplane, passing through its center $\bar{x}$, is contained in another ellipsoid,

$$\bar{E} \subseteq E_+ \coloneqq \mathcal{E}(\bar{x}_+, G_+), \tag{2.7.13}$$

with a sufficiently smaller relative volume:

$$\mathrm{vol}(E_+/E) \leq \exp\bigl(-1/(2n)\bigr). \tag{2.7.14}$$

The ellipsoid E_+ is given by[23]

$$\bar{x}_+ \coloneqq \bar{x} - \frac{1}{n+1}\frac{G^{-1}g}{\|g\|_G^*}, \tag{2.7.15}$$

$$G_+ \coloneqq \frac{n^2-1}{n^2}\Bigl(G + \frac{2}{n-1}\frac{gg^*}{(\|g\|_G^*)^2}\Bigr). \tag{2.7.16}$$

Proof. To simplify the computations, we can assume without loss of generality that $\|g\|_G^* = 1$. Let $x \in \mathbb{E}$ be arbitrary. Using (2.7.16), we obtain

$$\|\bar{x}_+ - x\|_{G_+}^2 = \frac{n^2-1}{n^2}\Bigl(\|\bar{x}_+ - x\|_G^2 + \frac{2}{n-1}\langle g, \bar{x}_+ - x\rangle^2\Bigr).$$

[23] Hereinafter, we assume that $n \geq 2$.

Further, in view of (2.7.15),

$$\|\bar{x}_+ - x\|_G^2 = \|\bar{x} - x\|_G^2 - \frac{2}{n+1}\langle g, \bar{x} - x\rangle + \frac{1}{(n+1)^2},$$
$$\langle g, \bar{x}_+ - x\rangle^2 = \langle g, \bar{x} - x\rangle^2 - \frac{2}{n+1}\langle g, \bar{x} - x\rangle + \frac{1}{(n+1)^2}.$$

Thus,

$$\|\bar{x}_+ - x\|_{G_+}^2 = \frac{n^2-1}{n^2}\Big(\|\bar{x} - x\|_G^2 - \frac{2}{n-1}\langle g, \bar{x} - x\rangle + \frac{2}{n-1}\langle g, \bar{x} - x\rangle^2 + \frac{1}{n^2-1}\Big).$$

Now assume that $x \in \bar{E}$. Then, $\|\bar{x} - x\|_G \le 1$ and $\langle g, \bar{x} - x\rangle \ge 0$. In particular, $\langle g, \bar{x} - x\rangle \le 1$ and

$$-\langle g, \bar{x} - x\rangle + \langle g, \bar{x} - x\rangle^2 = -\langle g, \bar{x} - x\rangle(1 - \langle g, \bar{x} - x\rangle) \le 0.$$

Thus, for any $x \in \bar{E}$, we have

$$\|\bar{x}_+ - x\|_{G_+}^2 \le \frac{n^2-1}{n^2}\Big(1 + \frac{1}{n^2-1}\Big) = 1.$$

This proves the inclusion in (2.7.13).

Let us prove (2.7.14). Using first Proposition 2.1.8 and then applying Propositions 2.1.4(ii) and 2.1.5 to the representation (2.7.16), we obtain

$$[\mathrm{vol}(E_+/E)]^{-1/2} = \det(G^{-1}, G_+) = \Big(\frac{n^2-1}{n^2}\Big)^n \frac{n+1}{n-1}$$
$$= \Big(\frac{n^2-1}{n^2}\Big)^{n-1}\Big(\frac{n+1}{n}\Big)^2 = \Big(1 - \frac{1}{n^2}\Big)^{n-1}\Big(1 + \frac{1}{n}\Big)^2.$$

It remains to show that $[\mathrm{vol}(E_+/E)]^{-1/2} \ge \exp(1/n)$. For this, it suffices to prove that, for any $\alpha \in (0,1)$, we have

$$\xi(\alpha) := (\alpha^{-1} - 1)\ln(1 - \alpha^2) + 2\ln(1 + \alpha) \ge \alpha. \tag{2.7.17}$$

But this is simple. Indeed, differentiating, we find that, for any $\alpha \in (0,1)$,

$$\xi'(\alpha) = -\alpha^{-2}\ln(1 - \alpha^2) - \frac{2\alpha(\alpha^{-1} - 1)}{1 - \alpha^2} + \frac{2}{1 + \alpha}$$

$$= -\alpha^{-2} \ln(1 - \alpha^2) \geq 1,$$

where the inequality follows from the concavity of the logarithm. This proves (2.7.17) since $\xi(\alpha) \to 0$ as $\alpha \to 0$. □

A more involved argument shows that the ellipsoid E_+ in Lemma 2.7.3 is, in fact, *optimal*: among all ellipsoids, containing $\bar{E}$, the ellipsoid E_+ has the smallest relative volume.

To finish the description of the Ellipsoid Method, it remains to specify how to choose the initial ellipsoid. For this, let us fix, as usual, some sufficiently simple operator $B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ and use it define the Euclidean norm in the space $\mathbb{E}$,

$$\|x\| \coloneqq \|x\|_B \coloneqq \langle Bx, x \rangle^{1/2}, \qquad x \in \mathbb{E}, \tag{2.7.18}$$

and the corresponding system of Euclidean balls:

$$B(\bar{x}, r) \coloneqq \{x \in \mathbb{E} : \|x - \bar{x}\| \leq r\}, \qquad \bar{x} \in \mathbb{E},\ r > 0. \tag{2.7.19}$$

Now, given any initial point $x_0 \in \mathbb{E}$, we can choose the initial ellipsoid as a Euclidean ball $B(x_0, R)$ of a sufficiently large radius R, such that it covers the whole feasible set Q.

Thus, we come to the following explicit scheme of the Ellipsoid Method.

Algorithm 2.7.2: Ellipsoid Method

Input: $x_0 \in \mathbb{E}$ and $R > 0$ such that $Q \subseteq B(x_0, R)$.

Initialization: Set $H_0 \coloneqq R^2 B^{-1}$.

Iteration $k \geq 0$**:**

1. Query the oracle to obtain $g_k \coloneqq \mathcal{G}(x_k)$.
2. Compute the center of the new ellipsoid:
$$x_{k+1} \coloneqq x_k - \frac{1}{n+1} \frac{H_k g_k}{\langle g_k, H_k g_k \rangle^{1/2}}.$$
3. Compute the operator of the new ellipsoid:
$$H_{k+1} \coloneqq \frac{n^2}{n^2 - 1} \Big(H_k - \frac{2}{n+1} \frac{H_k g_k g_k^* H_k}{\langle g_k, H_k g_k \rangle} \Big).$$

Remark 2.7.4. Note that the rules for updating centers and operators at Steps 2 and 3 of Algorithm 2.7.2 are exactly the same as the rules, given by

(2.7.15) and (2.7.16), with the only difference that, instead of updating the "primal" operators G_k, we directly update their inverses $H_k \equiv G_k^{-1}$ to keep the iteration cost at the level of $O(n^2)$.

Let us present an efficiency estimate for the Ellipsoid Method. Similarly to the general cutting plane scheme, the approximate solution to problem (2.7.1), produced by Algorithm 2.7.2 after $k \geq 1$ iterations, is defined as the best among all feasible search points, generated so far:

$$x_k^* := \operatorname{argmin}\{f(x) : x \in \{x_0, \ldots, x_{k-1}\} \cap \operatorname{int} Q\}. \tag{2.7.20}$$

If all points $x_0, \ldots, x_{k-1}$ are infeasible ($\notin \operatorname{int} Q$), we leave x_k^* undefined.

Theorem 2.7.5. *Consider iteration $k \geq 1$ of Algorithm 2.7.2. Suppose*

$$\delta_k := \exp\bigl(-k/(2n^2)\bigr)\frac{R}{r} < 1,$$

where $r > 0$ is the largest of the radii of Euclidean balls (of the form (2.7.19)) contained in Q. Then, the approximate solution (2.7.20) is well-defined, and

$$f(x_k^*) - f^* \leq \delta_k V.$$

Proof. In view of (2.7.12), (2.7.18) and (2.7.19), $B(x_0, R) = \mathcal{E}(x_0, R^{-2}B)$. Combining this observation with Lemma 2.7.3 and Remark 2.7.4, we conclude that Algorithm 2.7.2 is an instance of the general cutting plane scheme (Algorithm 2.7.1) with the localizers $\Omega_k := \mathcal{E}(x_k, H_k^{-1})$ satisfying

$$\operatorname{vol}(\Omega_{k+1}/\Omega_k) \leq \exp\bigl(-1/(2n)\bigr) \tag{2.7.21}$$

for all $k \geq 0$.

Let avrad be the average radius size function from (2.7.8), defined w.r.t. the solid $B_0 := \Omega_0$. Let $k \geq 1$ be arbitrary. Then,

$$\delta_k' := \frac{\operatorname{avrad}\Omega_k}{\operatorname{avrad} Q} = \left[\frac{\operatorname{vol}(\Omega_k/\Omega_0)}{\operatorname{vol}(Q/\Omega_0)}\right]^{1/n}. \tag{2.7.22}$$

From (2.7.21), we obtain, using Proposition 2.1.7(v), that

$$\operatorname{vol}(\Omega_k/\Omega_0) = \prod_{i=0}^{k-1} \operatorname{vol}(\Omega_{i+1}/\Omega_i) \leq \exp\bigl(-k/(2n)\bigr). \tag{2.7.23}$$

Now let us estimate $\operatorname{vol}(Q/\Omega_0)$ from below. By our assumptions, there

exists $\bar{x} \in \mathbb{E}$ such that

$$B_r := B(\bar{x}, r) \equiv \mathcal{E}(\bar{x}, r^{-2}B) \subseteq Q.$$

Therefore, by Propositions 2.1.7(ii), 2.1.8, 2.1.4(ii) and 2.1.4(i), we have

$$\operatorname{vol}(Q/\Omega_0) \geq \operatorname{vol}(B_r/\Omega_0) = [\det(r^2 B^{-1}, R^{-2}B)]^{1/2} = \Big(\frac{r}{R}\Big)^n. \tag{2.7.24}$$

Substituting (2.7.23) and (2.7.24) into (2.7.22), we obtain

$$\delta'_k \leq \exp\big(-k/(2n^2)\big)\frac{r}{R} = \delta_k.$$

It remains to apply Theorem 2.7.2 with the avrad size function. □

According to Theorem 2.7.5, for generating an ε-approximate solution (in terms of function value) to problem (2.7.1), the Ellipsoid Method requires at most the following number of iterations:

$$2n^2 \ln \frac{RV}{r\varepsilon} \tag{2.7.25}$$

(provided that $0 < \varepsilon < V$). The main factor in complexity estimate (2.7.25) is the dimensionality of the space. All other problem parameters enter this estimate under the logarithm.

Note that, in principle, we can always estimate the variation V (defined in (2.7.3)) from above using the Lipschitz constant $M > 0$ of f on the set Q and the simple bound on the diameter of Q: $\operatorname{diam} Q \leq 2R$ (which follows from the fact that $Q \subseteq B(x_0, R)$). Then, we obtain the following estimate:

$$2n^2 \ln \frac{2MR^2}{r\varepsilon}. \tag{2.7.26}$$

Comparing estimate (2.7.26) with the corresponding estimate (2.6.13) for the Subgradient Method, we see that the Ellipsoid Method has a much better dependency on the target accuracy ε, Lipschitz constant M, and the diameter of the feasible set $D := 2R$. However, we also see that the complexity estimate of the Ellipsoid Method grows *unboundedly* with the dimensionality of the space n, which does not happen with the Subgradient Method. In other words, the Ellipsoid Method does not, in general, have better guarantees than the Subgradient Method (even in terms of the number of oracle calls, not considering arithmetical complexities), which seems

to be rather strange. We will return to this issue and address it in more detail in Chapter 5, where we will also consider the Ellipsoid Method in a broader context of solving general problems with convex structure, such as saddle-point problems, variational inequalities, etc.

Chapter 3

Classical Quasi-Newton Methods

One of the main theoretical results about classical quasi-Newton methods for smooth optimization is their local *superlinear convergence*. Specifically, it is known that the ratio of successive residuals[1] r_k in these methods tends to zero as the iteration counter k goes to infinity:

$$\lim_{k\to\infty} \frac{r_{k+1}}{r_k} = 0.$$

Nevertheless, it is important that this result is only *qualitative*. It simply states that the superlinear convergence will eventually occur without specifying neither the *rate* of this superlinear convergence, nor its *starting moment*. In particular, it is unknown how exactly these quantities depend on the parameters of the problem. It is therefore important to obtain some *explicit* inequalities, describing the superlinear convergence of quasi-Newton methods.

In this chapter, we address this problem. We consider classical quasi-Newton methods from the convex Broyden class, which includes the most popular BFGS and DFP algorithms. For these methods, we present some explicit and non-asymptotic bounds on the rate of their local superlinear convergence under the standard assumption that the objective function is strongly convex with Lipschitz continuous gradient and Hessian. The main

[1] This could be the distance from the current iterate x_k to the optimum x^*, or the norm of the gradient $\nabla f(x_k)$.

parameters in our estimates are the problem dimension and its condition number.

Contents

The outline of this chapter is as follows. First, in Section 3.1, we study the convex Broyden class and establish a number of important properties of quasi-Newton updates from this class. Then, in Section 3.2, we analyze the standard quasi-Newton scheme, based on the updating rules from the convex Broyden class, as applied to minimizing a quadratic function. On this simple example, where the Hessian is constant, we illustrate the main ideas of our analysis. To extend the analysis onto more general nonlinear functions, we first introduce, in Section 3.3, the definition of a *strongly self-concordant* function and study some of its properties. The new definition provides us with a convenient affine-invariant alternative to the standard assumption of the Lipschitz continuous Hessian. In Section 3.4, we consider the general nonlinear unconstrained optimization problem and the corresponding classical quasi-Newton scheme for solving it. We show that, for this scheme, it is possible to prove absolutely the same results as for the quadratic function, provided that the starting point is sufficiently good.

This chapter mainly follows [160], and contains several auxiliary results that were originally presented in [161]. Apart from some minor changes in notation, we have additionally made a few small modifications, compared to [160]. First, we have introduced a new term "operator-revealing form" to distinguish between the special representation of quasi-Newton updates, used in this chapter, and the classical one from Section 2.5. We have also clarified the equivalence of the two representations. Second, in Section 3.2, we have added the comparison of the efficiency estimates of BFGS and DFP. Finally, we have included a new Section 3.3 containing the definition of strongly self-concordant functions and their main properties that were first presented in [159]. We have also added new Lemma 3.3.3 and Proposition 3.3.4 together with the accompanying discussion.

3.1 Convex Broyden Class

In this section, we establish several important properties of quasi-Newton updates from the convex Broyden class, which will be needed in our convergence analysis.

Let us start by introducing our notation. Everywhere in this chapter, it will be convenient for us to represent quasi-Newton updates in the special *operator-revealing form*. Specifically, given two positive definite operators $A, G \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ and a direction $u \in \mathbb{E} \setminus \{0\}$, we define the *BFGS* and *DFP updates* of G w.r.t. A along u by, respectively,

$$\mathrm{BFGS}(A, G, u) \coloneqq G - \frac{Guu^*G}{\langle Gu, u\rangle} + \frac{Auu^*A}{\langle Au, u\rangle}, \tag{3.1.1}$$

$$\mathrm{DFP}(A, G, u) \coloneqq G - \frac{Auu^*G + Guu^*A}{\langle Au, u\rangle} + \left(\frac{\langle Gu, u\rangle}{\langle Au, u\rangle} + 1\right)\frac{Auu^*A}{\langle Au, u\rangle}. \tag{3.1.2}$$

Comparing these formulas with the ones from (2.5.30) and (2.5.24), we see that they indeed correspond to the standard BFGS and DFP updates, which we discussed in Section 2.5, for $\delta = u$ and $\gamma = Au$. Note that, in the classical BFGS and DFP methods for minimizing a twice differentiable strongly convex function f, at each iteration, given the current iterate x and the new one x_+, we choose u as their difference and A as the integral Hessian on the segment $[x, x_+]$:

$$u = x_+ - x, \qquad A \coloneqq \int_0^1 \nabla^2 f(x + tu)dt.$$

Of course, for implementing the corresponding BFGS and DFP updates, we do not actually need to compute A as we only need the product

$$Au = \nabla f(x_+) - \nabla f(x).$$

The representation of an update in the operator-revealing form is more convenient than the standard one when we need to explicitly emphasize the *target operator* A which we are trying to approximate by doing the update.

The *Broyden class* of quasi-Newton updates is defined as follows: for any $A, G \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$, $u \in \mathbb{E} \setminus \{0\}$ and $\tau \in \mathbb{R}$, we set

$$\mathrm{Broyd}_\tau(A, G, u) \coloneqq (1 - \varphi_\tau)\,\mathrm{BFGS}(A, G, u) + \varphi_\tau\,\mathrm{DFP}(A, G, u), \tag{3.1.3}$$

where $\varphi_\tau \coloneqq \varphi_\tau(A, G, u)$ is the following linear fractional function of τ:

$$\varphi_\tau \coloneqq \tau\frac{\langle Au, u\rangle}{\langle AG^{-1}Au, u\rangle}\left[\tau\frac{\langle Au, u\rangle}{\langle AG^{-1}Au, u\rangle} + (1 - \tau)\frac{\langle Gu, u\rangle}{\langle Au, u\rangle}\right]^{-1}. \tag{3.1.4}$$

In the case when the expression in the brackets in (3.1.4) is zero, we leave

both φ_τ and $\mathrm{Broyd}_\tau(A, G, u)$ undefined. For the sake of convenience, we also define $\mathrm{Broyd}_\tau(A, G, u) \coloneqq G$ when $u = 0$.

It is not difficult to see that (3.1.3) and (3.1.4) is just an alternative *dual parametrization* of the standard Broyden class (2.5.37), which we introduced in Section 2.5.2. Specifically, the transformation, defined in (3.1.4), is a bijection, and its range is the whole real line except, possibly, one singular point φ_∞ (corresponding to $\tau = \pm\infty$), for which the Broyden update (3.1.3) results in a degenerate operator. For us, the dual parametrization will be more convenient as it corresponds to a simple linear parametrization of the inverse update.

Lemma 3.1.1. *Let $A, G \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$, $u \in \mathbb{E} \setminus \{0\}$ and $\tau \in \mathbb{R}$ be such that $G_+ \coloneqq \mathrm{Broyd}_\tau(A, G, u)$ is well-defined. Then, G_+ is invertible, and*

$$G_+^{-1} = (1-\tau)\,\mathrm{BFGS}^{-1}(A, G, u) + \tau\,\mathrm{DFP}^{-1}(A, G, u), \tag{3.1.5}$$

$$\det(G_+^{-1}, G) = (1-\tau)\frac{\langle Gu, u\rangle}{\langle Au, u\rangle} + \tau\frac{\langle Au, u\rangle}{\langle AG^{-1}Au, u\rangle}, \tag{3.1.6}$$

where

$$\begin{aligned}\mathrm{BFGS}^{-1}(A, G, u) \coloneqq G^{-1} &- \frac{G^{-1}Auu^* + uu^*AG^{-1}}{\langle Au, u\rangle} \\ &+ \Big(\frac{\langle AG^{-1}Au, u\rangle}{\langle Au, u\rangle} + 1\Big)\frac{uu^*}{\langle Au, u\rangle},\end{aligned} \tag{3.1.7}$$

$$\mathrm{DFP}^{-1}(A, G, u) \coloneqq G^{-1} - \frac{G^{-1}Auu^*AG^{-1}}{\langle AG^{-1}Au, u\rangle} + \frac{uu^*}{\langle Au, u\rangle}. \tag{3.1.8}$$

Proof. See Section 3.A.1. □

Remark 3.1.2. Formulas (3.1.7) and (3.1.8) are exactly the inverse BFGS and DFP updates from (2.5.31) and (2.5.26), respectively, written in the operator-revealing form.

In this chapter, we will be interested only in the *convex* Broyden class, which is described by the values of $\tau \in [0, 1]$. Note that, for all such τ, the expression in the brackets in (3.1.4) is always positive for any $u \neq 0$, so both φ_τ and $\mathrm{Broyd}_\tau(A, G, u)$ are well-defined; moreover, $\varphi_\tau \in [0, 1]$. The two extreme members of this class, corresponding to the values of $\tau = 0$ and $\tau = 1$, are the BFGS and DFP updates, respectively.

A basic property of an update from the convex Broyden class—a *convex Broyden update*—is that it preserves the bounds on the eigenvalues w.r.t.

the target operator.

Lemma 3.1.3. *Let $A, G \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ and $\xi, \eta \geq 1$ be such that*

$$\xi^{-1} A \preceq G \preceq \eta A.$$

Then, for any $u \in \mathbb{E}$ and any $\tau \in [0, 1]$, we have

$$\xi^{-1} A \preceq \mathrm{Broyd}_\tau(A, G, u) \preceq \eta A. \tag{3.1.9}$$

Proof. We can assume that $u \neq 0$ since otherwise the claim is trivial. Since $\varphi_\tau \in [0, 1]$ in (3.1.3), it suffices to prove (3.1.9) only for the DFP and BFGS updates independently.

Note that the DFP update can be written in the following form:

$$\mathrm{DFP}(A, G, u) = \Big(I_{\mathbb{E}^*} - \frac{Auu^*}{\langle Au, u\rangle}\Big) G \Big(I_{\mathbb{E}} - \frac{uu^* A}{\langle Au, u\rangle}\Big) + \frac{Auu^* A}{\langle Au, u\rangle},$$

where $I_{\mathbb{E}}$, $I_{\mathbb{E}^*}$ are the identity operators in the spaces $\mathbb{E}$, $\mathbb{E}^*$, respectively. From this representation, it follows that

$$\begin{aligned}
\mathrm{DFP}(A, G, u) &\preceq \eta \Big(I_{\mathbb{E}^*} - \frac{Auu^*}{\langle Au, u\rangle}\Big) A \Big(I_{\mathbb{E}} - \frac{uu^* A}{\langle Au, u\rangle}\Big) + \frac{Auu^* A}{\langle Au, u\rangle} \\
&= \eta \Big(A - \frac{Auu^* A}{\langle Au, u\rangle}\Big) + \frac{Auu^* A}{\langle Au, u\rangle} \\
&= \eta A - (\eta - 1) \frac{Auu^* A}{\langle Au, u\rangle} \preceq \eta A, \\
\mathrm{DFP}(A, G, u) &\succeq \xi^{-1} \Big(I_{\mathbb{E}^*} - \frac{Auu^*}{\langle Au, u\rangle}\Big) A \Big(I_{\mathbb{E}} - \frac{uu^* A}{\langle Au, u\rangle}\Big) + \frac{Auu^* A}{\langle Au, u\rangle} \\
&= \xi^{-1} \Big(A - \frac{Auu^* A}{\langle Au, u\rangle}\Big) + \frac{Auu^* A}{\langle Au, u\rangle} \\
&= \xi^{-1} A + (1 - \xi^{-1}) \frac{Auu^* A}{\langle Au, u\rangle} \succeq \xi^{-1} A.
\end{aligned}$$

For the BFGS update, we apply Lemma 3.A.1:

$$\begin{aligned}
\mathrm{BFGS}(A, G, u) &= G - \frac{Guu^* G}{\langle Gu, u\rangle} + \frac{Auu^* A}{\langle Au, u\rangle} \\
&\preceq \eta \Big(A - \frac{Auu^* A}{\langle Au, u\rangle}\Big) + \frac{Auu^* A}{\langle Au, u\rangle} \\
&= \eta A - (\eta - 1) \frac{Auu^* A}{\langle Au, u\rangle} \preceq \eta A,
\end{aligned}$$

$$\begin{aligned}\mathrm{BFGS}(A,G,u) &= G - \frac{Guu^*G}{\langle Gu,u\rangle} + \frac{Auu^*A}{\langle Au,u\rangle} \\ &\succeq \xi^{-1}\Big(A - \frac{Auu^*A}{\langle Au,u\rangle}\Big) + \frac{Auu^*A}{\langle Au,u\rangle} \\ &= \xi^{-1}A + (1-\xi^{-1})\frac{Auu^*A}{\langle Au,u\rangle} \succeq \xi^{-1}A.\end{aligned}$$

Combining the above results, we obtain the claim. □

Remark 3.1.4. Lemma 3.1.3 was first established by Fletcher [59] in a slightly stronger form and using a different argument. He also demonstrated that one of the relations in (3.1.9) may no longer be valid when the Broyden update is not convex.

Let us define, for any $A, G \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ and $u \in \mathbb{E} \setminus \{0\}$, the following measure of closeness of G to A along the direction u:

$$\nu(A,G,u) := \frac{\|(G-A)u\|_G^*}{\|u\|_A}. \tag{3.1.10}$$

Our next goal is to show that, by iterating convex Broyden updates, we can force the measure ν to converge to zero. For this, we will study how certain potential functions change after one convex Broyden update, and estimate the corresponding improvement from below by a certain non-negative monotonically increasing function of ν, vanishing at zero.

First, let us consider the *log-det barrier* potential function[2]:

$$V(A,G) := \ln\det(A^{-1}, G), \tag{3.1.11}$$

defined for any $A, G \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$. It will be useful only when we can guarantee that $A \preceq G$, and hence $V(A,G) \geq 0$.

Lemma 3.1.5. *Let $A, G \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ and $\eta \geq 1$ be such that*

$$A \preceq G \preceq \eta A.$$

Then, for any $\tau \in [0,1]$, $u \in \mathbb{E} \setminus \{0\}$ and $G_+ := \mathrm{Broyd}_\tau(A,G,u)$, we have

$$V(A,G) - V(A,G_+) \geq \ln\big(1 + (\tau\eta^{-1} + 1 - \tau)\nu^2(A,G,u)\big). \tag{3.1.12}$$

[2] Recall that $\det(\cdot,\cdot)$ is the determinant product defined in (2.1.31).

Proof. Using Lemma 3.1.1, we obtain

$$\begin{aligned} V(A,G) - V(A,G_+) &= \ln\det(G_+^{-1}, G) \\ &= \ln\left(\tau\frac{\langle Au,u\rangle}{\langle AG^{-1}Au,u\rangle} + (1-\tau)\frac{\langle Gu,u\rangle}{\langle Au,u\rangle}\right) \\ &= \ln\left(1+\tau\frac{\langle A(A^{-1}-G^{-1})Au,u\rangle}{\langle AG^{-1}Au,u\rangle} + (1-\tau)\frac{\langle (G-A)u,u\rangle}{\langle Au,u\rangle}\right). \end{aligned} \tag{3.1.13}$$

Since[3] $0 \preceq G - A \preceq (1-\eta^{-1})G$, we have

$$\begin{aligned} (G-A)G^{-1}(G-A) &\preceq (1-\eta^{-1})(G-A) \\ &\preceq (1+\eta^{-1})^{-1}(G-A) \preceq G - A. \end{aligned} \tag{3.1.14}$$

Therefore, denoting $\nu \coloneqq \nu(A,G,u)$, we can write that

$$\frac{\langle (G-A)u,u\rangle}{\langle Au,u\rangle} \overset{(3.1.14)}{\geq} \frac{\langle (G-A)G^{-1}(G-A)u,u\rangle}{\langle Au,u\rangle} \overset{(3.1.10)}{=} \nu^2,$$

and, since $A(A^{-1}-G^{-1})A = G - A - (G-A)G^{-1}(G-A)$, that

$$\begin{aligned} \frac{\langle A(A^{-1}-G^{-1})Au,u\rangle}{\langle AG^{-1}Au,u\rangle} &= \frac{\langle (G-A-(G-A)G^{-1}(G-A))u,u\rangle}{\langle AG^{-1}Au,u\rangle} \\ &\overset{(3.1.14)}{\geq} \eta^{-1}\frac{\langle (G-A)G^{-1}(G-A)u,u\rangle}{\langle AG^{-1}Au,u\rangle} \\ &\geq \eta^{-1}\frac{\langle (G-A)G^{-1}(G-A)u,u\rangle}{\langle Au,u\rangle} \\ &\overset{(3.1.10)}{=} \eta^{-1}\nu^2. \end{aligned}$$

Substituting these two inequalities into (3.1.13), we obtain (3.1.12). □

Now consider another potential function, the *augmented log-det barrier*:

$$\psi(G,A) \coloneqq \ln\det(A^{-1},G) - \langle G^{-1}, G-A\rangle \quad (\geq 0), \tag{3.1.15}$$

defined for any $A, G \in \mathcal{S}_{++}(\mathbb{E},\mathbb{E}^*)$. Note that this function is, in fact, the *Bregman divergence* generated by the log-det prox function (see (2.5.15), (2.5.27) and (2.5.29)). Therefore, $\psi(G,A)$ is indeed nonnegative for any $A, G \in \mathcal{S}_{++}(\mathbb{E},\mathbb{E}^*)$. As a result, this potential function is more universal

[3] This is obvious when $G-A$ is non-degenerate. The general case follows by continuity.

than the previous one as we can work with it even when the condition $A \preceq G$ is violated.

Remark 3.1.6. It is worth mentioning that, instead of $\psi(G, A)$, it is possible to study the evolution of $\psi(A, G)$, or, in other words, to center the Bregman divergence at the target operator A instead of its current approximation G. This is indeed a reasonable approach. However, as it turns out, in the end, it leads to slower rates of superlinear convergence (for more details, see [161]).

In order to relate the improvement in our new potential function ψ with the directional measure of closeness ν, we need an auxiliary inequality.

Lemma 3.1.7. *For any real $\alpha \geq \beta > 0$, we have $\alpha + \beta^{-1} - 1 \geq 1$, and*

$$\alpha - \ln \beta - 1 \geq \frac{\sqrt{3}}{2+\sqrt{3}} \ln(\alpha + \beta^{-1} - 1) \geq \frac{6}{13} \ln(\alpha + \beta^{-1} - 1). \tag{3.1.16}$$

Proof. We only need to prove the first inequality in (3.1.16) since the second one follows from it and the fact that

$$\frac{\sqrt{3}+2}{\sqrt{3}} = 1 + \frac{2}{\sqrt{3}} \leq 1 + \frac{7}{6} = \frac{13}{6}.$$

Let $\beta > 0$ be fixed, and let $\zeta_1 \colon (1 - \beta^{-1}, +\infty) \to \mathbb{R}$ be the function

$$\zeta_1(\alpha) \coloneqq \alpha - \frac{\sqrt{3}}{2+\sqrt{3}} \ln(\alpha + \beta^{-1} - 1).$$

Note that the domain of ζ_1 includes the point $\alpha = \beta$ since $\beta \geq 2 - \beta^{-1} > 1 - \beta^{-1}$. Let us show that ζ_1 increases on the interval $[\beta, +\infty)$. Indeed, for any $\alpha \geq \beta$, we have

$$\begin{aligned} \zeta_1'(\alpha) &= 1 - \frac{\sqrt{3}}{2+\sqrt{3}} \frac{1}{\alpha + \beta^{-1} - 1} \\ &> 1 - \frac{1}{\alpha + \beta^{-1} - 1} = \frac{\alpha + \beta^{-1} - 2}{\alpha + \beta^{-1} - 1} \geq \frac{\beta + \beta^{-1} - 2}{\alpha + \beta^{-1} - 1} \geq 0. \end{aligned}$$

Thus, it is sufficient to prove (3.1.16) only in the case when $\alpha = \beta$. Equivalently, we need to show that the function $\zeta_2 \colon (0, +\infty) \to \mathbb{R}$ defined by

$$\zeta_2(\alpha) \coloneqq \alpha - \ln \alpha - 1 - \frac{\sqrt{3}}{2+\sqrt{3}} \ln(\alpha + \alpha^{-1} - 1)$$

is nonnegative. Differentiating, we find that, for all $\alpha > 0$, we have

$$\begin{aligned}
\zeta_2'(\alpha) &= 1 - \alpha^{-1} - \frac{\sqrt{3}}{2+\sqrt{3}} \frac{1-\alpha^{-2}}{\alpha + \alpha^{-1} - 1} \\
&= (1-\alpha^{-1}) \left(1 - \frac{\sqrt{3}}{2+\sqrt{3}} \frac{1+\alpha^{-1}}{\alpha + \alpha^{-1} - 1} \right) \\
&= (1-\alpha^{-1}) \frac{\alpha + \alpha^{-1} - 1 - (2\sqrt{3}-3)(1+\alpha^{-1})}{\alpha + \alpha^{-1} - 1} \\
&= (1-\alpha^{-1}) \frac{\alpha - 2(\sqrt{3}-1) + (\sqrt{3}-1)^2 \alpha^{-1}}{1 + \alpha^{-1} - 1} \\
&= (1-\alpha^{-1}) \frac{(\sqrt{\alpha} - (\sqrt{3}-1)/\sqrt{\alpha})^2}{\alpha + \alpha^{-1} - 1}.
\end{aligned}$$

Hence, $\zeta_2'(\alpha) \leq 0$ for $0 < \alpha \leq 1$, and $\zeta_2'(\alpha) \geq 0$ for $\alpha \geq 1$. Thus, the minimum of ζ_2 is attained at $\alpha = 1$. Consequently, $\zeta_2(\alpha) \geq \zeta_2(1) = 0$ for all $\alpha > 0$. □

Using Lemma 3.1.7, we can now show that the improvement in the augmented log-det barrier potential function can be bounded from below by exactly the same logarithmic function of ν (up to an absolute constant), which we had for our first potential function.

Lemma 3.1.8. *Let $A, G \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ and $\xi, \eta \geq 1$ be such that*

$$\xi^{-1} A \preceq G \preceq \eta A.$$

Then, for any $\tau \in [0,1]$, $u \in \mathbb{E} \setminus \{0\}$ and $G_+ := \mathrm{Broyd}_\tau(A, G, u)$, we have

$$\psi(G, A) - \psi(G_+, A) \geq \frac{6}{13} \ln\bigl(1 + (\tau[\xi\eta]^{-1} + 1 - \tau)\nu^2(A, G, u)\bigr).$$

Proof. According to Lemma 3.1.1, we have

$$\begin{aligned}
&\langle G^{-1} - G_+^{-1}, A \rangle \\
&\quad = \tau \left[\frac{\langle AG^{-1}AG^{-1}Au, u \rangle}{\langle AG^{-1}Au, u \rangle} - 1 \right] + (1-\tau) \left[\frac{\langle AG^{-1}Au, u \rangle}{\langle Au, u \rangle} - 1 \right], \\
&\qquad \det(G_+^{-1}, G) = \tau \frac{\langle Au, u \rangle}{\langle AG^{-1}Au, u \rangle} + (1-\tau) \frac{\langle Gu, u \rangle}{\langle Au, u \rangle}.
\end{aligned}$$

Thus, in view of (3.1.15),

$$\begin{aligned}\psi(G,A) - \psi(G_+,A) &= \langle G^{-1} - G_+^{-1}, A\rangle + \ln\det(G_+^{-1}, G) \\ &= \tau\alpha_1 + (1-\tau)\alpha_0 + \ln\bigl(\tau\beta_1^{-1} + (1-\tau)\beta_0^{-1}\bigr) - 1 \\ &= \alpha - \ln\beta - 1,\end{aligned} \tag{3.1.17}$$

where we denote

$$\begin{aligned}\alpha_1 &:= \frac{\langle AG^{-1}AG^{-1}Au, u\rangle}{\langle AG^{-1}Au, u\rangle}, \qquad \beta_1 := \frac{\langle AG^{-1}Au, u\rangle}{\langle Au, u\rangle}, \\ \alpha_0 &:= \frac{\langle AG^{-1}Au, u\rangle}{\langle Au, u\rangle}, \qquad \beta_0 := \frac{\langle Au, u\rangle}{\langle Gu, u\rangle}, \\ \alpha &:= \tau\alpha_1 + (1-\tau)\alpha_0, \qquad \beta := (\tau\beta_1^{-1} + (1-\tau)\beta_0^{-1})^{-1}.\end{aligned}$$

Note that $\alpha_1 \geq \beta_1$ and $\alpha_0 \geq \beta_0$ by the Cauchy-Schwartz inequality. At the same time, $\tau\beta_1 + (1-\tau)\beta_2 \geq \beta$ by the convexity of the inverse function $t \mapsto t^{-1}$. Hence, we can apply Lemma 3.1.7 to estimate the right-hand side in (3.1.17) from below. It remains to note that

$$\begin{aligned}\alpha + \beta^{-1} - 1 &= \tau\frac{\langle (A + AG^{-1}AG^{-1}A)u, u\rangle}{\langle AG^{-1}Au, u\rangle} + (1-\tau)\frac{\langle (AG^{-1}A + G)u, u\rangle}{\langle Au, u\rangle} - 1 \\ &= 1 + \tau\frac{\langle (G-A)G^{-1}AG^{-1}(G-A)\rangle}{\langle AG^{-1}Au, u\rangle} \\ &\quad + (1-\tau)\frac{\langle (G-A)G^{-1}(G-A)u, u\rangle}{\langle Au, u\rangle} \\ &\geq 1 + (\tau[\xi\eta]^{-1} + 1 - \tau)\frac{\langle (G-A)G^{-1}(G-A)u, u\rangle}{\langle Au, u\rangle} \\ &= 1 + (\tau[\xi\eta]^{-1} + 1 - \tau)\nu^2(A, G, u).\end{aligned}$$

Putting everything together, we obtain the claim. □

The measure ν, defined in (3.1.10), is the ratio of the norm of $(G-A)u$ measured w.r.t. G, and the norm of u measured w.r.t. A. Let us show how we can change the corresponding norms to those, induced by G_+ and G, respectively.

Lemma 3.1.9. *Let $A, G \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ and $\xi > 0$ be such that*

$$\xi^{-1} A \preceq G. \tag{3.1.18}$$

Then, for any $\tau \in [0,1]$, $u \in \mathbb{E} \setminus \{0\}$, and $G_+ := \mathrm{Broyd}_\tau(A, G, u)$, we have

$$\nu(A, G, u) \geq (1+\xi)^{-1/2} \frac{\|(G-A)u\|^*_{G_+}}{\|u\|_G}.$$

Proof. From (3.1.5), it is easy to see that $G_+^{-1} A u = u$. Hence,

$$\begin{aligned} &\frac{\langle (G-A) G_+^{-1} (G-A) u, u \rangle}{\langle Gu, u \rangle} \\ &\quad = \frac{\langle G G_+^{-1} G u, u \rangle}{\langle Gu, u \rangle} + \frac{\langle Au, G_+^{-1} Au \rangle}{\langle Gu, u \rangle} - 2 \frac{\langle Gu, G_+^{-1} Au \rangle}{\langle Gu, u \rangle} \\ &\quad = \frac{\langle G G_+^{-1} G u, u \rangle}{\langle Gu, u \rangle} + \frac{\langle Au, u \rangle}{\langle Gu, u \rangle} - 2. \end{aligned} \tag{3.1.19}$$

Since $1 - t \leq t^{-1} - 1$ for all $t > 0$, we further have, in view of (3.1.5),

$$\begin{aligned} \frac{\langle G G_+^{-1} G u, u \rangle}{\langle Gu, u \rangle} &= \tau \left[1 - \frac{\langle Au, u \rangle^2}{\langle Gu, u \rangle \langle AG^{-1} Au, u \rangle} + \frac{\langle Gu, u \rangle}{\langle Au, u \rangle} \right] \\ &\quad + (1-\tau) \left[\left(\frac{\langle AG^{-1} Au, u \rangle}{\langle Au, u \rangle} + 1 \right) \frac{\langle Gu, u \rangle}{\langle Au, u \rangle} - 1 \right] \\ &\leq \left(\frac{\langle AG^{-1} Au, u \rangle}{\langle Au, u \rangle} + 1 \right) \frac{\langle Gu, u \rangle}{\langle Au, u \rangle} - 1. \end{aligned} \tag{3.1.20}$$

Denote $\nu := \nu(A, G, u)$. According to (3.1.10),

$$\nu^2 = \frac{\langle (G-A) G^{-1} (G-A) u, u \rangle}{\langle Au, u \rangle} = \frac{\langle Gu, u \rangle}{\langle Au, u \rangle} + \frac{\langle AG^{-1} Au, u \rangle}{\langle Au, u \rangle} - 2. \tag{3.1.21}$$

Consequently, in view of (3.1.18), (3.1.21) and (3.1.20),

$$\begin{aligned} (1+\xi)\nu^2 &\geq \left(\frac{\langle AG^{-1} Au, u \rangle}{\langle Au, u \rangle} + 1 \right) \nu^2 \\ &= \left(\frac{\langle AG^{-1} Au, u \rangle}{\langle Au, u \rangle} + 1 \right) \frac{\langle Gu, u \rangle}{\langle Au, u \rangle} \\ &\quad + \frac{\langle AG^{-1} Au, u \rangle^2}{\langle Au, u \rangle^2} - \frac{\langle AG^{-1} Au, u \rangle}{\langle Au, u \rangle} - 2 \end{aligned}$$

$$\geq \frac{\langle GG_+^{-1}Gu, u\rangle}{\langle Au, u\rangle} + \frac{\langle AG^{-1}Au, u\rangle^2}{\langle Au, u\rangle} - \frac{\langle AG^{-1}Au, u\rangle}{\langle Au, u\rangle} - 1,$$

Combining this with (3.1.19), we obtain

$$\begin{aligned}
&(1+\xi)\nu^2 - \frac{\langle (G-A)G_+^{-1}(G-A)u, u\rangle}{\langle Gu, u\rangle} \\
&\quad = (1+\xi)\nu^2 - \frac{\langle GG_+^{-1}Gu, u\rangle}{Gu, u\rangle} - \frac{\langle Au, u\rangle}{\langle Gu, u\rangle} + 2 \\
&\quad \geq \frac{\langle AG^{-1}Au, u\rangle^2}{\langle Au, u\rangle^2} - \frac{\langle AG^{-1}Au, u\rangle}{\langle Au, u\rangle} - \frac{\langle Au, u\rangle}{\langle Gu, u\rangle} + 1 \\
&\quad \geq \frac{\langle AG^{-1}Au, u\rangle^2}{\langle Au, u\rangle^2} - 2\frac{\langle AG^{-1}Au, u\rangle}{\langle Au, u\rangle} + 1 \geq 0,
\end{aligned}$$

where we the penultimate inequality is due to the Cauchy–Schwartz inequality $\frac{\langle Au,u\rangle}{\langle Gu,u\rangle} \leq \frac{\langle AG^{-1}Au,u\rangle}{\langle Au,u\rangle}$. □

3.2 Unconstrained Quadratic Minimization

Let us study the convergence properties of the classical quasi-Newton methods from the convex Broyden class, as applied to minimizing the following strongly convex quadratic function:

$$f(x) := \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle, \tag{3.2.1}$$

where $A \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ and $b \in \mathbb{E}^*$.

Let us fix some operator $B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ which will be used for initializing the methods. Denote by $\mu > 0$ the strong convexity parameter of f, and by $L > 0$ the Lipschitz constant of the gradient of f, both measured w.r.t. the Euclidean norm, induced by B (see Proposition 2.2.5(iii) and Corollary 2.2.9):

$$\mu B \preceq A \preceq LB. \tag{3.2.2}$$

The corresponding *condition number* is defined as usual:

$$\varkappa := \frac{L}{\mu} \ (\geq 1). \tag{3.2.3}$$

Consider the following standard quasi-Newton scheme for minimizing the quadratic function (3.2.1). For the sake of simplicity, we assume that

the constant L is known.

Algorithm 3.2.1: Convex Broyden Method for Quadratic Function

Initialization: Choose $x_0 \in \mathbb{E}$. Set $G_0 = LB$.

For $k \geq 0$ iterate:

1. Update $x_{k+1} = x_k - G_k^{-1} \nabla f(x_k)$.
2. Set $u_k = x_{k+1} - x_k$ and choose $\tau_k \in [0, 1]$.
3. Compute $G_{k+1} = \text{Broyd}_{\tau_k}(A, G_k, u_k)$.

Remark 3.2.1. We present Algorithm 3.2.1 in a rather specific form which is convenient for its theoretical analysis. In an actual implementation of this method, in order to keep the iteration cost at the level of $O(n^2)$, instead of the Hessian approximations G_k, it is typical to work directly with their inverses $H_k \coloneqq G_k^{-1}$. The corresponding update of H_k into H_{k+1} can be efficiently implemented in $O(n^2)$ operations using the formulas from Lemma 3.1.1.

Note that Algorithm 3.2.1 starts with $G_0 = LB$. Therefore, its first iteration is identical to that of the standard Gradient Method with constant step size (see Algorithm 2.3.1):

$$x_1 = x_0 - \frac{1}{L} B^{-1} \nabla f(x_0).$$

For measuring the rate of convergence of Algorithm 3.2.1, it will be convenient to use the norm of the gradient, induced by the Hessian:

$$\lambda_k \coloneqq \|\nabla f(x_k)\|_A^* = \langle \nabla f(x_k), A^{-1} \nabla f(x_k) \rangle^{1/2}, \qquad k \geq 0.$$

This measure of optimality is directly related to the functional residual. Indeed, denoting by $x^* = A^{-1} b$ the minimizer of f, and by f^* the corresponding minimal value, we obtain, for any $k \geq 0$,

$$\begin{aligned} f(x_k) - f^* &= \frac{1}{2} \langle A(x_k - x^*), x_k - x^* \rangle = \frac{1}{2} \langle Ax_k - b, A^{-1}(Ax_k - b) \rangle \\ &= \frac{1}{2} \langle \nabla f(x_k), A^{-1} \nabla f(x_k) \rangle = \frac{1}{2} \lambda_k^2. \end{aligned}$$

Let us show that Algorithm 3.2.1 has global linear convergence, and that the corresponding rate is at least as good as that of the standard Gradient

Method (cf. Theorem 2.3.1).

Theorem 3.2.2. *In Algorithm 3.2.1, for all $k \geq 0$, we have*

$$A \preceq G_k \preceq \varkappa A, \tag{3.2.4}$$

$$\lambda_k \leq (1 - \varkappa^{-1})^k \lambda_0. \tag{3.2.5}$$

Proof. For $k = 0$, (3.2.4) follows from the fact that $G_0 = LB$ and (3.2.2). For all other $k \geq 1$, it can be justified by induction using Lemma 3.1.3.

Let us prove (3.2.5). Let $k \geq 0$ be arbitrary. By the definitions of u_k and x_{k+1} in Algorithm 3.2.1, and the fact that f is a quadratic function with Hessian A, we have

$$G_k u_k = -\nabla f(x_k), \qquad A u_k = \nabla f(x_{k+1}) - \nabla f(x_k).$$

Hence,

$$\begin{aligned} \lambda_k^2 &= \langle G_k A^{-1} G_k u_k, u_k \rangle, \\ \lambda_{k+1}^2 &= \langle (G_k - A) A^{-1} (G_k - A) u_k, u_k \rangle. \end{aligned} \tag{3.2.6}$$

Note, from (3.2.4), that

$$0 \preceq A^{-1} - G_k^{-1} \preceq (1 - \varkappa^{-1}) A^{-1}.$$

Therefore,

$$\begin{aligned} (G_k - A) A^{-1} (G_k - A) &= G_k (A^{-1} - G_k^{-1}) A (A^{-1} - G_k^{-1}) G_k \\ &\preceq (1 - \varkappa^{-1})^2 G_k A^{-1} G_k. \end{aligned}$$

Consequently, according to (3.2.6),

$$\lambda_{k+1} \leq (1 - \varkappa^{-1}) \lambda_k.$$

This proves (3.2.5) since $k \geq 0$ was arbitrary. □

Now let us establish the *superlinear* convergence of Algorithm 3.2.1. According to Theorem 3.2.2, for the quadratic function, we have $A \preceq G_k$ for all $k \geq 0$. Therefore, in our analysis, we can use both potential functions: the log-det barrier and the augmented log-det barrier. Let us consider both options. We start with the first one.

In what follows, for each $k \geq 1$, we denote by $\varkappa_k$ the following transformation of the original condition number $\varkappa$, corresponding to the first k

iterations of Algorithm 3.2.1:

$$\varkappa_k \coloneqq \prod_{i=0}^{k-1} (\tau_i \varkappa^{-1} + 1 - \tau_i)^{-1/k} \quad (\geq 1). \tag{3.2.7}$$

Recall that $\tau_i \in [0,1]$ are the "parameters" of Algorithm 3.2.1 responsible for the choice of the particular "type" of the Hessian approximation update. For the two most important examples, BFGS and DFP, we have $\tau_i \equiv 0$ and $\tau_i \equiv 1$, respectively, and thus

$$\varkappa_k^{\mathrm{BFGS}} \equiv 1, \qquad \varkappa_k^{\mathrm{DFP}} \equiv \varkappa. \tag{3.2.8}$$

In general, however, both τ_k and $\varkappa_k$ are allowed to change at each iteration k (although we do not really explore this possibility any further).

Theorem 3.2.3. *In Algorithm 3.2.1, for all $k \geq 1$, we have*

$$\lambda_k \leq \bigl[2\varkappa_k\bigl(\varkappa^{n/k} - 1\bigr)\bigr]^{k/2} \sqrt{\varkappa}\, \lambda_0. \tag{3.2.9}$$

Proof. Without loss of generality, we can assume that $u_i \neq 0$ for all $0 \leq i \leq k$. Denote $V_i \coloneqq V(A, G_i) \geq 0$, $\nu_i \coloneqq \nu(A, G_i, u_i)$, $p_i \coloneqq \tau_i \varkappa^{-1} + 1 - \tau_i$, $g_i \coloneqq \|\nabla f(x_i)\|_{G_i}^*$ for any $0 \leq i \leq k$. By Lemma 3.1.5 and (3.2.4), for all $0 \leq i \leq k-1$, we have

$$\ln(1 + p_i \nu_i^2) \leq V_i - V_{i+1}.$$

Summing up these inequalities for all $0 \leq i \leq k-1$, we obtain

$$\begin{aligned} \sum_{i=0}^{k-1} \ln(1 + p_k \nu_k^2) &\leq V_0 - V_k \leq V_0 = V(A, LB) \\ &\overset{(3.1.11)}{=} \ln\det(A^{-1}, LB) \overset{(3.2.2)}{\leq} \ln\det\bigl((\mu B)^{-1}, LB\bigr) = n \ln \varkappa. \end{aligned} \tag{3.2.10}$$

Hence, by the convexity of function $t \mapsto \ln(1 + e^t)$, from (3.2.10), we get

$$\begin{aligned} \frac{n \ln \varkappa}{k} &\geq \frac{1}{k} \sum_{i=0}^{k-1} \ln(1 + p_i \nu_i^2) = \frac{1}{k} \sum_{i=0}^{k-1} \ln\bigl(1 + \exp\{\ln(p_i \nu_i^2)\}\bigr) \\ &\geq \ln\biggl(1 + \exp\Bigl\{\frac{1}{k} \sum_{i=0}^{k-1} \ln(p_i \nu_i^2)\Bigr\}\biggr) = \ln\Bigl(1 + \Bigl[\prod_{i=0}^{k-1} p_i \nu_i^2\Bigr]^{1/k}\Bigr). \end{aligned} \tag{3.2.11}$$

But, for all $0 \leq i \leq k-1$, we have

$$\nu_i^2 \geq 2^{-1} \frac{\langle (G_i - A)G_{i+1}^{-1}(G_i - A)u_i, u_i \rangle}{\langle G_i u_i, u_i \rangle} = 2^{-1} \frac{g_{i+1}^2}{g_i^2}$$

in view of Lemma 3.1.9 and (3.2.4), and since $G_i u_i = -\nabla f(x_i)$ while $Au_i = \nabla f(x_{i+1}) - \nabla f(x_i)$. Hence,

$$\prod_{i=0}^{k-1} \nu_i^2 \geq 2^{-k} \frac{g_k^2}{g_0^2},$$

and so, from (3.2.11), it follows that

$$\frac{n \ln \varkappa}{k} \geq \ln \left(1 + (2\varkappa_k)^{-1} \left[\frac{g_k}{g_0} \right]^{2/k} \right),$$

where $\varkappa_k := \prod_{i=0}^{k-1} p_i^{-1/k}$. Rearranging, we obtain

$$g_k \leq \left[2\varkappa_k \left(\varkappa^{n/k} - 1 \right) \right]^{k/2} g_0.$$

It remains to note that $\lambda_k \leq \sqrt{\varkappa}\, g_k$ and $g_0 \leq \lambda_0$ in view of (3.2.4). □

Remark 3.2.4. As can be seen from (3.2.10), the factor $n \ln \varkappa$ in (3.2.9) can be improved up to $\ln \det(A^{-1}, LB) = \sum_{i=1}^n \ln(L/\lambda_i)$, where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of A w.r.t. B (see Proposition 2.1.4(vi)). This improved factor can be significantly smaller than the original one if the majority of the eigenvalues λ_i are much larger than μ.

Let us discuss the efficiency estimate from Theorem 3.2.3. Note that its *minimal* value over all $\tau_i \in [0, 1]$ is achieved at $\tau_i \equiv 0$ (see (3.2.7)). This corresponds to the *BFGS Method*, for which we have, according to (3.2.8),

$$\lambda_k \leq \left[2\left(\varkappa^{n/k} - 1 \right) \right]^{k/2} \sqrt{\varkappa}\, \lambda_0. \tag{3.2.12}$$

Although this bound is formally valid for all $k \geq 1$, it becomes useful[4] only when the expression in front of λ_0 in the right-hand side of (3.2.12) is less or equal than 1. The smallest integer $k = \hat{K}_0^{\mathrm{BFGS}} \geq 1$ for which this happens can be thought of as the *starting moment* of the superlinear convergence of

[4] Indeed, from Theorem 3.2.2, we know that $\lambda_k \leq \lambda_0$ for all $k \geq 0$.

the BFGS Method, according to estimate (3.2.12). Let us show that[5]

$$\hat{K}_0^{\mathrm{BFGS}} \sim n \ln \varkappa,$$

or, more precisely, that

$$\bar{K}_0^{\mathrm{BFGS}} := \lceil 2n \ln \varkappa \rceil \leq \hat{K}_0^{\mathrm{BFGS}} \leq \lceil 4n \ln \varkappa \rceil =: K_0^{\mathrm{BFGS}}. \tag{3.2.13}$$

Indeed, since $\exp(t) > 1 + t$ for any $t > 0$, we have, for all $1 \leq k < \bar{K}_0^{\mathrm{BFGS}}$,

$$2(\varkappa^{n/k} - 1) = 2\bigl(\exp([n \ln \varkappa]/k) - 1\bigr) > 2\frac{n \ln \varkappa}{k} \geq 1.$$

This proves that $\hat{K}_0^{\mathrm{BFGS}} \geq \bar{K}_0^{\mathrm{BFGS}}$. On the other hand, using the inequality $\exp(t) \leq (1-t)^{-1} = 1 + t/(1-t)$, which is valid for any $t < 1$, we obtain, for all $k \geq K_0^{\mathrm{BFGS}}$,

$$\exp([n \ln \varkappa]/k) - 1 \leq \frac{(n \ln \varkappa)/k}{1 - (n \ln \varkappa)/k} \leq \frac{4}{3}\frac{n \ln \varkappa}{k}.$$

Further, for all $k \geq K_0^{\mathrm{BFGS}}$,

$$\sqrt{\varkappa} = \exp\Bigl(\frac{1}{2} \ln \varkappa\Bigr) \leq \exp\Bigl(\frac{1}{8}k\Bigr) = \Bigl[\exp\Bigl(\frac{1}{4}\Bigr)\Bigr]^{k/2} \leq \Bigl(\frac{3}{2}\Bigr)^{k/2}.$$

Combining these inequalities with (3.2.12), we obtain, for all $k \geq K_0^{\mathrm{BFGS}}$,

$$\lambda_k \leq \Bigl(\frac{8}{3}\frac{n \ln \varkappa}{k}\Bigr)^{k/2} \sqrt{\varkappa}\, \lambda_0 \leq \Bigl(4\frac{n \ln \varkappa}{k}\Bigr)^{k/2} \lambda_0 \ (\leq \lambda_0). \tag{3.2.14}$$

This proves that $\hat{K}_0^{\mathrm{BFGS}} \leq K_0^{\mathrm{BFGS}}$.

In contrast, the *maximal* value of the efficiency estimate from Theorem 3.2.3 over all $\tau_i \in [0, 1]$ is achieved at $\tau_i \equiv 1$ (see (3.2.7)). This corresponds to the *DFP Method*, for which we have, according to (3.2.8),

$$\lambda_k \leq \bigl[2\varkappa\bigl(\varkappa^{n/k} - 1\bigr)\bigr]^{k/2} \sqrt{\varkappa}\, \lambda_0 \tag{3.2.15}$$

for all $k \geq 1$. Repeating the same reasoning as above, we can easily obtain that the starting moment $\hat{K}_0^{\mathrm{DFP}}$ of the superlinear convergence of DFP,

[5] Hereinafter, we assume that $\mu < L$. Otherwise, in view of Theorem 3.2.2, the method finds the exact solution after one iteration.

according to (3.2.15), is

$$\hat{K}_0^{\mathrm{DFP}} \sim n\varkappa \ln \varkappa,$$

or, more precisely,

$$\bar{K}_0^{\mathrm{DFP}} := \lceil 2n\varkappa \ln \varkappa \rceil \leq \hat{K}_0^{\mathrm{DFP}} \leq \lceil 4n\varkappa \ln \varkappa \rceil =: K_0^{\mathrm{DFP}}.$$

In particular, for all $k \geq K_0^{\mathrm{DFP}}$, we have

$$\lambda_k \leq \Big(4\frac{n\varkappa \ln \varkappa}{k}\Big)^{k/2} \lambda_0 \ (\leq \lambda_0).$$

According to our estimates, the BFGS Method is almost insensitive to the condition number $\varkappa$ since this quantity enters the principal efficiency estimates (3.2.13) and (3.2.14) *under the logarithm*. The DFP Method, on the contrary, is very sensitive to the condition number. Compared to BFGS, its superlinear convergence begins $\varkappa$ times later, and the corresponding rate is much slower.

Let us briefly present another approach for justifying the superlinear convergence rate of the form (3.2.9) for Algorithm 3.2.1. This approach is based on our second potential function, namely, the *augmented* log-det barrier.

Theorem 3.2.5. *In Algorithm 3.2.1, for all $k \geq 1$, we have*

$$\lambda_k \leq \big[2\varkappa_k\big(\varkappa^{13n/(6k)} - 1\big)\big]^{k/2} \sqrt{\varkappa}\, \lambda_0, \tag{3.2.16}$$

where $\varkappa_k$ is defined in (3.2.7).

Proof. Without loss of generality, we can assume that $u_i \neq 0$ for all $0 \leq i \leq k$. Denote $\psi_i := \psi(G_i, A)$, $\nu_i := \nu(A, G_i, u_i)$ and $p_i := \tau_i \varkappa^{-1} + 1 - \tau_i$ for all $0 \leq i \leq k$. By Lemma 3.1.8 and (3.2.4), for all $0 \leq i \leq k-1$, we have

$$\frac{6}{13} \ln(1 + p_i \nu_i^2) \leq \psi_i - \psi_{i+1}.$$

Summing up these inequalities for $0 \leq i \leq k-1$, we obtain

$$\begin{aligned} \frac{6}{13} \sum_{i=0}^{k-1} \ln(1 + p_i \nu_i^2) \leq \psi_0 - \psi_k &\overset{(3.1.15)}{\leq} \psi_0 = \psi(LB, A) \\ &\overset{(3.1.15)}{=} \ln \det(A^{-1}, LB) - \langle (LB)^{-1}, LB - A \rangle \\ &\overset{(3.2.2)}{\leq} n \ln \varkappa. \end{aligned} \tag{3.2.17}$$

Now we can continue exactly as in the proof of Theorem 3.2.3. □

Comparing our new efficiency estimate (3.2.16) with the previous one from (3.2.9), we see that they differ only in an absolute constant under the exponent. Thus, for the quadratic function, we do not gain anything by working with the augmented potential function instead of the usual one. Nevertheless, our second proof turns out to be more universal and, in contrast to the first one, can be extended onto general nonlinear functions, as we will see in Section 3.4. But first let us introduce a certain assumption on the objective function, which will be convenient in our future analysis.

3.3 Strongly Self-Concordant Functions

Traditionally, the convergence analysis of quasi-Newton methods for general nonlinear minimization problems is done under the assumptions that the objective function is strongly convex and has Lipschitz continuous gradient and Hessian (see, e.g., Theorems 2.5.8 and 2.5.9). However, as we already discussed in Section 2.4.1, the corresponding constants in all these assumptions are, in general, not affine-invariant, since they depend on the particular norm, which we use for measuring them. As a result, the classical strong convexity and Lipschitz continuity assumptions are badly suited for the analysis of affine-invariant methods such as Newton's Method.

As we already know from Section 2.4.1, for Newton's Method, a better assumption about the objective function is that of self-concordance. Since quasi-Newton methods aim at approximating Newton's Method, it is therefore reasonable to use something similar for them as well. Unfortunately, the ideal goal of having only affine-invariant constants in the analysis of quasi-Newton methods is, in general, unreachable since, in these methods, there is an initial Hessian approximation which, in principle, can be arbitrary. Nevertheless, as we will see, it is possible to get a little closer to our ideal goal. Specifically, we can replace one of the three classical assumptions, namely, the Lipschitz continuity of the Hessian, with a certain self-concordance assumption in which the constant is affine-invariant.

Definition 3.3.1 (Strongly self-concordant function)**.** A function $f : \mathbb{E} \to \mathbb{R}$ is called *strongly self-concordant* if it is twice differentiable with strictly positive definite Hessian, and there exists a constant $M \geq 0$ (parameter of strong self-concordance) such that, for all $x, y, z, w \in \mathbb{E}$,

$$\nabla^2 f(x) - \nabla^2 f(y) \preceq M \|x - y\|_z \, \nabla^2 f(w), \tag{3.3.1}$$

where $\|\cdot\|_z$ is the norm induced by $\nabla^2 f(z)$.

Note that strongly self-concordant functions form a subclass of self-concordant functions. Indeed, let us choose arbitrarily a point $x \in \mathbb{E}$, direction $h \in \mathbb{E}$, and scalar $t > 0$. Then, from (3.3.1), it follows that

$$\nabla^2 f(x + th) - \nabla^2 f(x) \preceq Mt\|h\|_x \nabla^2 f(x).$$

Hence, according to (2.1.12) and (2.1.14),

$$\langle [\nabla^2 f(x + th) - \nabla^2 f(x)]h, h\rangle \leq Mt\|h\|_x^3.$$

Dividing this inequality by t and computing the limit as $t \to 0$, we obtain

$$D^3 f(x)[h]^3 \leq M\|h\|_x^3.$$

Thus, f is self-concordant with constant $M/2$ (see Definition 2.4.3).

The simplest example of a strongly self-concordant function is a strictly convex quadratic function, for which we have $M = 0$. A more general example is given by a strongly convex function with Lipschitz continuous Hessian (cf. Lemma 2.4.4).

Lemma 3.3.2. *Let $f : \mathbb{E} \to \mathbb{R}$ be a μ-strongly convex function with L_2-Lipschitz continuous Hessian, where both constants $\mu > 0$ and $L_2 \geq 0$ are measured w.r.t. a certain Euclidean norm. Then, f is strongly self-concordant with parameter*

$$M = \frac{L_2}{\mu^{3/2}}.$$

Proof. We assume that the constants μ and L_2 are measured w.r.t. to the Euclidean norm $\|\cdot\| := \|\cdot\|_B$ with $B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$. According to Proposition 2.2.5(iii), for all $x \in \mathbb{E}$, we have

$$\nabla^2 f(x) \succeq \mu B.$$

Combining this inequality with the Lipschitz continuity of the Hessian, we obtain, for all $x, y, z, w \in \mathbb{E}$,

$$\begin{aligned}\nabla^2 f(x) - \nabla^2 f(y) &\preceq L_2\|x - y\|B = L_2\langle B(x - y), x - y\rangle^{1/2} B \\ &\preceq \frac{L_2}{\mu^{3/2}}\|x - y\|_z \nabla^2 f(w),\end{aligned}$$

and the claim follows. □

Lemma 3.3.2 shows that, under the extra assumption of strong convexity, the Lipschitz continuity of the Hessian implies strong self-concordance. It turns out that the reverse implication is also true but under a different extra assumption, namely, the Lipschitz continuity of the gradient.

Lemma 3.3.3. *Let $f\colon \mathbb{E} \to \mathbb{R}$ be a strongly self-concordant function with parameter $M \geq 0$. Suppose that the gradient of f is Lipschitz continuous with constant $L \geq 0$ (w.r.t. a certain Euclidean norm). Then, the Hessian of f is also Lipschitz continuous (w.r.t. the same norm) with constant*

$$L_2 = ML^{3/2}.$$

Proof. We assume that the constant L is measured w.r.t. to the Euclidean norm $\|\cdot\| \coloneqq \|\cdot\|_B$ with $B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$. According to Proposition 2.2.8, for all $x \in \mathbb{E}$, we have

$$\nabla^2 f(x) \preceq LB.$$

Combining this with (3.3.1), we obtain, for all $x, y \in \mathbb{E}$,

$$\begin{aligned}\nabla^2 f(x) - \nabla^2 f(y) &\preceq M\langle \nabla^2 f(x)(x-y), x-y\rangle^{1/2} \nabla^2 f(x) \\ &\preceq ML^{3/2}\|x-y\|B.\end{aligned}$$

The claim follows. □

Putting Lemmas 3.3.2 and 3.3.3 together, we see that, under the extra assumptions of strong convexity and Lipschitz continuity of the gradient, the strong self-concordance is, in fact, equivalent to the Lipschitz continuity of the Hessian. In other words, the following statement holds.

Proposition 3.3.4. *The class of strongly convex and strongly self-concordant functions with Lipschitz continuous gradient is exactly the same as the class of strongly convex functions with Lipschitz continuous gradient and Hessian.*

According to Proposition 3.3.4, on the *qualitative* level, studying the traditional class of strongly convex functions with Lipschitz continuous gradient and Hessian is equivalent to studying the class of strongly convex and strongly self-concordant functions with Lipschitz gradient. However, on the *quantitative* level, the latter class may be better to work with, since the strong self-concordance parameter is affine-invariant, in contrast to the Lipschitz constant of the Hessian[6].

[6] See also Section 4.5.1 for an example of a function with a "small" strong self-concordance parameter M but a "large" Lipschitz constant L_2 (w.r.t. the standard norm).

Let us conclude this section by establishing several simple relations between the Hessians of a strongly self-concordant function, which will be useful in our subsequent analysis.

Lemma 3.3.5. *Let $f\colon \mathbb{E} \to \mathbb{R}$ be a strongly self-concordant function with parameter $M \geq 0$, and let $x, y \in \mathbb{E}$. Denote $r \coloneqq \|y - x\|_x$. Then,*

$$(1 + Mr)^{-1}\nabla^2 f(x) \preceq \nabla^2 f(y) \preceq (1 + Mr)\nabla^2 f(x). \tag{3.3.2}$$

Further, for $J \coloneqq \int_0^1 \nabla^2 f\big(x + t(y - x)\big)dt$ and any $v \in \{x, y\}$, we have

$$(1 + \tfrac{1}{2}Mr)^{-1}\nabla^2 f(v) \preceq J \preceq (1 + \tfrac{1}{2}Mr)\nabla^2 f(v). \tag{3.3.3}$$

Proof. Denote $h \coloneqq y - x$. Taking $z = w = x$ in (3.3.1), we obtain

$$\nabla^2 f(y) - \nabla^2 f(x) \preceq Mr\nabla^2 f(x),$$

which gives us the second relation in (3.3.2) after moving $\nabla^2 f(x)$ into the right-hand side. Interchanging now x and y in (3.3.1) and taking $z = x$, $w = y$, we get

$$\nabla^2 f(x) - \nabla^2 f(y) \preceq Mr\nabla^2 f(y),$$

which gives us the first relation in (3.3.2) after moving $\nabla^2 f(x)$ into the right-hand side and then dividing by $1 + Mr$.

Let us now prove (3.3.3) for $v = x$ (the proof for $v = y$ is similar). Choosing $y = x + th$ in (3.3.1) for $t > 0$, and $w = z = x$, we obtain

$$\nabla^2 f(x + th) - \nabla^2 f(x) \preceq M\|th\|_x \nabla^2 f(x) = Mrt\nabla^2 f(x).$$

This proves the second relation in (3.3.3) after integrating for t from 0 to 1 and moving $\nabla^2 f(x)$ into the right-hand side. Interchanging x and y in (3.3.1) and taking $y = x + th$ for $t > 0$, $z = x$, while leaving w arbitrary, we get

$$\nabla^2 f(x) - \nabla^2 f(x + th) \preceq M\|-th\|_x \nabla^2 f(w) = Mrt\nabla^2 f(w).$$

Hence, by integrating for t from 0 to 1, we see that

$$\nabla^2 f(x) - J \preceq \tfrac{1}{2}Mr\nabla^2 f(w).$$

Taking now $w = x + th$ and integrating again, we obtain

$$\nabla^2 f(x) - J \preceq \tfrac{1}{2} Mr \int_0^1 \nabla^2 f(x + th) dt = \tfrac{1}{2} MrJ,$$

and the first inequality in (3.3.3) follows after moving J to the right-hand side and dividing by $1 + \frac{1}{2}Mr$. □

3.4 Minimization of General Functions

In this section, we consider the general unconstrained minimization problem:

$$\min_{x \in \mathbb{E}} f(x), \tag{3.4.1}$$

where $f : \mathbb{E} \to \mathbb{R}$ is a twice differentiable function. We assume that the function f is strongly convex, strongly self-concordant and its gradient is Lipschitz continuous, i.e., there exist $\mu, L > 0$ and $M \geq 0$ such that, for all $x, y, z, w \in \mathbb{E}$, it holds

$$\mu B \preceq \nabla^2 f(x) \preceq LB, \tag{3.4.2}$$

$$\nabla^2 f(x) - \nabla^2 f(y) \preceq M \|x - y\|_z \nabla^2 f(w), \tag{3.4.3}$$

where $B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ is a certain fixed operator. For convenience, we also introduce the following *condition number* of problem (3.4.1):

$$\varkappa := \frac{L}{\mu} \ (\geq 1). \tag{3.4.4}$$

Remark 3.4.1. Since we are mostly interested in *local* convergence guarantees, it is possible to relax our assumptions by requiring that (3.4.2) and (3.4.3) hold only in a certain neighborhood of a solution x^* to problem (3.4.1). For this, it suffices to assume that the Hessian of f is Lipschitz continuous in this neighborhood, and $\nabla^2 f(x^*)$ is nonsingular, which are exactly the standard assumptions used in many classical works on local convergence of quasi-Newton methods (see, e.g., [46]). However, to avoid excessive technicalities, we do not do this.

Our goal is to study the convergence properties of the following standard quasi-Newton scheme for solving problem (3.4.1).

Algorithm 3.4.1: Convex Broyden Method
Initialization: Choose $x_0 \in \mathbb{E}$. Set $G_0 = LB$.
For $k \geq 0$ **iterate:** 1. Update $x_{k+1} = x_k - G_k^{-1} \nabla f(x_k)$. 2. Set $u_k = x_{k+1} - x_k$ and choose $\tau_k \in [0, 1]$. 3. Denote $J_k = \int_0^1 \nabla^2 f(x_k + t u_k) dt$. 4. Set $G_{k+1} = \mathrm{Broyd}_{\tau_k}(J_k, G_k, u_k)$.

Remark 3.4.2. Similarly to Remark 3.2.1, we present Algorithm 3.4.1 in a rather specific form which is convenient for its theoretical analysis. When implementing this method, it is common to work directly with the inverse Hessian approximations $H_k := G_k^{-1}$ instead of G_k in order to keep the iteration cost at the level of $O(n^2)$. Also, note that it is not necessary to compute the integral Hessian J_k explicitly since, for implementing the corresponding inverse Hessian approximation update at Step 4, one only needs the product

$$J_k u_k = \nabla f(x_{k+1}) - \nabla f(x_k),$$

which is just the difference of two successive gradients.

Remark 3.4.3. Recall that the operator B is allowed to be arbitrary. Therefore, in principle, instead of setting $G_0 = LB$ in Algorithm 3.4.1 and assuming (3.4.2) and (3.4.4), we could equivalently say that the initial Hessian approximation G_0 is an arbitrary positive definite linear operator such that

$$\varkappa^{-1} G_0 \preceq \nabla^2 f(x) \preceq G_0$$

for some $\varkappa \geq 1$ and all $x \in \mathbb{E}$. This actually corresponds to measuring the parameter of strong convexity and the Lipschitz constant of the gradient of f w.r.t. the norm $\|\cdot\|_{G_0}$, with $\varkappa$ being the corresponding condition number. However, we prefer to work in terms of B in order to keep the notation consistent and draw some parallels with the Gradient Method (Algorithm 2.3.1).

For measuring the convergence rate of Algorithm 3.4.1, we use the local gradient norm:

$$\lambda_k := \|\nabla f(x_k)\|_{x_k}^* = \langle \nabla f(x_k), [\nabla^2 f(x_k)]^{-1} \nabla f(x_k) \rangle^{1/2}. \tag{3.4.5}$$

The local convergence analysis of Algorithm 3.4.1 is, in general, the same as the corresponding analysis in the quadratic case. However, it is much more technical due to the fact that, in the nonlinear case, the Hessian is no longer constant. This causes a few problems.

First, there are now several different ways how one can treat the Hessian approximation G_k. One can view it as an approximation to the Hessian $\nabla^2 f(x_k)$ at the current iterate x_k, to the Hessian $\nabla^2 f(x^*)$ at the minimizer x^*, to the integral Hessian J_k etc. Of course, locally, due to strong self-concordance, all these variants are equivalent since the corresponding Hessians are close to each other. Nevertheless, from the viewpoint of technical simplicity of the analysis, some options are slightly more preferable than others. We find it to be the most convenient to always think of G_k as an approximation to the integral Hessian J_k.

The second issue is as follows. Suppose we already know the connection between our current Hessian approximation G_k and the actual integral Hessian J_k, e.g., in terms of the relative eigenvalues and the value of the augmented log-det barrier potential function (3.1.15). Naturally, we want to know how these quantities change after we update G_k into G_{k+1} at Step 4 of Algorithm 3.4.1. For this, we apply Lemma 3.1.3 and Lemma 3.1.8, respectively. However, the problem is that both of these lemmas will provide us only with the information on the connection between the update result G_{k+1} and the *current* integral Hessian J_k (which was used for performing the update), not the next one J_{k+1}. Therefore, we need to additionally take into account the errors, resulting from approximating J_{k+1} by J_k.

For estimating the errors, which accumulate as a result of approximating one Hessian by another, it is convenient to introduce the following quantities[7]:

$$r_k \coloneqq \|u_k\|_{x_k}, \quad \xi_k \coloneqq \exp\Big(M \sum_{i=0}^{k-1} r_i\Big) \ (\geq 1), \qquad k \geq 0, \tag{3.4.6}$$

where M is the constant of strong self-concordance from (3.4.3). The quantity ξ_k will be helpful for upper bounding various products of the form $\prod_{i=0}^{k-1}(1 + Mr_i)$. Despite the presence of the exponent in its definition, ξ_k will not actually be too large, as the sum $M \sum_{i=0}^{k-1} r_i$ turns out to be uniformly bounded by a certain "small" absolute constant whenever the initial point x_0 is sufficiently good.

[7] We follow the standard convention that the sum over the empty set is defined as 0, so $\xi_0 = 1$. Similarly, the product over the empty set is defined as 1.

We analyze Algorithm 3.4.1 in several steps. The first step is to establish the bounds on the relative eigenvalues of the Hessian approximations w.r.t. the corresponding Hessians.

Lemma 3.4.4. *For all $k \geq 0$, we have*

$$\xi_k^{-1} \nabla^2 f(x_k) \preceq G_k \preceq \xi_k \varkappa \nabla^2 f(x_k), \tag{3.4.7}$$

$$\xi_{k+1}^{-1} J_k \preceq G_k \preceq \xi_{k+1} \varkappa J_k. \tag{3.4.8}$$

Proof. For $k = 0$, (3.4.7) follows from (3.4.2) and the fact that $G_0 = LB$ and $\xi_0 = 1$. Now suppose that $k \geq 0$, and that (3.4.7) has already been proved for all indices up to k. Then, combining Lemma 3.3.5 with (3.4.7), we obtain

$$[\xi_k(1 + \tfrac{1}{2} M r_k)]^{-1} J_k \preceq G_k \preceq (1 + \tfrac{1}{2} M r_k) \xi_k \varkappa J_k. \tag{3.4.9}$$

Since $(1 + \frac{1}{2} M r_k)\xi_k \leq \xi_{k+1}$ by (3.4.6), this proves (3.4.8) for the index k. Applying Lemma 3.1.3 to (3.4.9), we get

$$[\xi_k(1 + \tfrac{1}{2} M r_k)]^{-1} J_k \preceq G_{k+1} \preceq (1 + \tfrac{1}{2} M r_k) \xi_k \varkappa J_k.$$

Combining this with Lemma 3.3.5 and (3.4.6), we obtain

$$G_{k+1} \preceq (1 + \tfrac{1}{2} M r_k)^2 \xi_k \varkappa \nabla^2 f(x_{k+1}) \preceq \xi_{k+1} \varkappa \nabla^2 f(x_{k+1}),$$
$$G_{k+1} \succeq [(1 + \tfrac{1}{2} M r_k)^2 \xi_k]^{-1} \nabla^2 f(x_{k+1}) \succeq \xi_{k+1}^{-1} \nabla^2 f(x_{k+1}).$$

This proves (3.4.7) for the index $k+1$, and we can continue by induction. □

Corollary 3.4.5. *For all $k \geq 0$, we have*

$$r_k \leq \xi_k \lambda_k. \tag{3.4.10}$$

Proof. Indeed, using the definitions of r_k, u_k and λ_k (see (3.4.6), Algorithm 3.4.1, and (3.4.5)), we obtain

$$r_k = \|u_k\|_{x_k} = \langle \nabla f(x_k), G_k^{-1} \nabla^2 f(x_k) G_k^{-1} \nabla f(x_k) \rangle^{1/2}$$
$$\leq \xi_k \langle \nabla f(x_k), \nabla^2 f(x_k)^{-1} \nabla f(x_k) \rangle^{1/2} = \xi_k \lambda_k,$$

where the inequality follows from (3.4.7). □

The second step in our analysis is to establish a preliminary version of

the linear convergence theorem for Algorithm 3.4.1.

Lemma 3.4.6. *For all $k \geq 0$, we have*

$$\lambda_k \leq \sqrt{\xi_k}\, \lambda_0 \prod_{i=0}^{k-1} q_i, \tag{3.4.11}$$

where

$$q_i := \max\{1 - (\xi_{i+1}\varkappa)^{-1}, \xi_{i+1} - 1\}. \tag{3.4.12}$$

Proof. Let $k, i \geq 0$ be arbitrary. By Taylor's formula and the definitions of J_i and u_i in Algorithm 3.4.1, we have

$$\nabla f(x_{i+1}) = \nabla f(x_i) + J_i u_i = J_i(J_i^{-1} - G_i^{-1})\nabla f(x_i).$$

Hence, in view of (3.4.5) and Lemma 3.3.5,

$$\begin{aligned} \lambda_{i+1}^2 &= \langle \nabla f(x_{i+1}), \nabla^2 f(x_{i+1})^{-1} \nabla f(x_{i+1}) \rangle \\ &\leq (1 + \tfrac{1}{2} M r_i) \langle \nabla f(x_{i+1}), J_i^{-1} \nabla f(x_{i+1}) \rangle \\ &\leq (1 + \tfrac{1}{2} M r_i) \langle \nabla f(x_i), (J_i^{-1} - G_i^{-1}) J_i (J_i^{-1} - G_i^{-1}) \nabla f(x_i) \rangle. \end{aligned} \tag{3.4.13}$$

According to (3.4.8),

$$-(\xi_{i+1} - 1) J_i^{-1} \preceq J_i^{-1} - G_i^{-1} \preceq [1 - (\xi_{i+1}\varkappa)^{-1}] J_i^{-1}.$$

Therefore, by (3.4.12) and Lemma 3.3.5,

$$(J_i^{-1} - G_i^{-1}) J_i (J_i^{-1} - G_i^{-1}) \preceq q_i^2 J_i^{-1} \preceq q_i^2 (1 + \tfrac{1}{2} M r_i) \nabla^2 f(x_i)^{-1}.$$

Thus, in view of (3.4.13) and (3.4.5), $\lambda_{i+1} \leq (1 + \frac{1}{2} M r_i) q_i \lambda_i$. Consequently,

$$\lambda_k \leq \lambda_0 \prod_{i=0}^{k-1} (1 + \tfrac{1}{2} M r_i) q_i \leq \lambda_0 \prod_{i=0}^{k-1} \exp(\tfrac{1}{2} M r_i) q_i \overset{(3.4.6)}{=} \sqrt{\xi_k}\, \lambda_0 \prod_{i=0}^{k-1} q_i. \quad \square$$

Next, we establish a preliminary version of the theorem on superlinear convergence of Algorithm 3.4.1. The proof uses the augmented log-det barrier potential function and is essentially a generalization of the corresponding proof of Theorem 3.2.5.

Lemma 3.4.7. *For all $k \geq 1$, we have*

$$\lambda_k \leq \left[(1 + \xi_k)\varkappa_k \big((\xi_{k+1}^{\xi_{k+1}} \varkappa)^{13n/(6k)} - 1 \big) \right]^{k/2} \sqrt{\xi_k \varkappa}\, \lambda_0, \tag{3.4.14}$$

where $\varkappa_k \coloneqq \prod_{i=0}^{k-1}(\tau_i \xi_{i+1}^{-2} \varkappa^{-1} + 1 - \tau_i)^{-1/k}$.

Proof. Without loss of generality, assume that $u_i \neq 0$ for all $0 \leq i \leq k$. Denote $\psi_i \coloneqq \psi(G_i, J_i)$, $\tilde{\psi}_{i+1} \coloneqq \psi(G_{i+1}, J_i)$, $\nu_i \coloneqq \nu(J_i, G_i, u_i)$, $p_i \coloneqq \tau_i \xi_{i+1}^{-2} \varkappa^{-1} + 1 - \tau_i$, and $g_i \coloneqq \|\nabla f(x_i)\|_{G_i}^*$ for any $0 \leq i \leq k$.

Let $0 \leq i \leq k-1$ be arbitrary. By Lemma 3.1.8 and (3.4.8), we have

$$\tfrac{6}{13} \ln\left(1 + p_i \nu_i^2\right) \leq \psi_i - \tilde{\psi}_{i+1} = \psi_i - \psi_{i+1} + \Delta_i, \tag{3.4.15}$$

where

$$\Delta_i \coloneqq \psi_{i+1} - \tilde{\psi}_{i+1} \overset{(3.1.15)}{=} \langle G_{i+1}^{-1}, J_{i+1} - J_i \rangle + \ln \det(J_{i+1}^{-1}, J_i). \tag{3.4.16}$$

Note that, in view of Lemma 3.3.5,

$$J_i \succeq (1 + \tfrac{1}{2} M r_i)^{-1} \nabla^2 f(x_{i+1}) \succeq (1 + \tfrac{1}{2} M r_i)^{-1} (1 + \tfrac{1}{2} M r_{i+1})^{-1} J_{i+1}.$$

In particular,

$$J_i \succeq \exp\bigl(-\tfrac{1}{2} M (r_i + r_{i+1})\bigr) J_{i+1} \succeq \bigl(1 - \tfrac{1}{2} M (r_i + r_{i+1})\bigr) J_{i+1}.$$

Combining this with (3.4.8) and (3.4.6), we obtain

$$\begin{aligned}
&\sum_{i=0}^{k-1} \langle G_{i+1}^{-1}, J_{i+1} - J_i \rangle \\
&\quad \leq \tfrac{1}{2} M \sum_{i=0}^{k-1} (r_i + r_{i+1}) \langle G_{i+1}^{-1}, J_{i+1} \rangle \leq \tfrac{1}{2} n M \sum_{i=0}^{k-1} \xi_{i+2} (r_i + r_{i+1}) \\
&\quad \leq \tfrac{1}{2} n M \xi_{k+1} \sum_{i=0}^{k-1} (r_i + r_{i+1}) \leq n M \xi_{k+1} \sum_{i=0}^{k} r_i = n \xi_{k+1} \ln \xi_{k+1}.
\end{aligned}$$

Consequently, according to (3.4.16),

$$\sum_{i=0}^{k-1} \Delta_i \leq n \xi_{k+1} \ln \xi_{k+1} + \ln \det(J_k^{-1}, J_0). \tag{3.4.17}$$

Summing up (3.4.15), we thus obtain

$$\begin{aligned}\frac{6}{13}\sum_{i=0}^{k-1}\ln(1+p_i\nu_i^2) &\le \psi_0-\psi_k+\sum_{i=0}^{k-1}\Delta_i \overset{(3.1.15)}{\le} \psi_0+\sum_{i=0}^{k-1}\Delta_i\\ &\overset{(3.1.15)}{=} \ln\det(J_0^{-1},LB)-\langle (LB)^{-1},LB-J_0\rangle+\sum_{i=0}^{k-1}\Delta_i\\ &\overset{(3.4.17)}{\le} \ln\det(J_k^{-1},LB)-\langle (LB)^{-1},LB-J_0\rangle+n\xi_{k+1}\ln\xi_{k+1}\\ &\overset{(3.4.2)}{\le} n\ln\varkappa+n\xi_{k+1}\ln\xi_{k+1}=n\ln(\xi_{k+1}^{\xi_{k+1}}\varkappa).\end{aligned}$$

By the convexity of the function $t\mapsto\ln(1+e^t)$, it follows that

$$\begin{aligned}&\frac{13}{6}\frac{n}{k}\ln(\xi_{k+1}^{\xi_{k+1}}\varkappa)\\ &\quad\ge \frac{1}{k}\sum_{i=0}^{k-1}\ln(1+p_i\nu_i^2)=\frac{1}{k}\sum_{i=0}^{k-1}\ln(1+\exp\{\ln(p_i\nu_i^2)\})\\ &\quad\ge \ln\Bigl(1+\exp\Bigl\{\frac{1}{k}\sum_{i=0}^{k-1}\ln(p_i\nu_i^2)\Bigr\}\Bigr)=\ln\Bigl(1+\Bigl[\prod_{i=0}^{k-1}p_i\nu_i^2\Bigr]^{1/k}\Bigr).\end{aligned}\tag{3.4.18}$$

At the same time,

$$\nu_i^2\ge(1+\xi_{i+1})^{-1}\frac{\langle (G_i-J_i)G_{i+1}^{-1}(G_i-J_i)u_i,u_i\rangle}{\langle G_iu_i,u_i\rangle}=(1+\xi_{i+1})^{-1}\frac{g_{i+1}^2}{g_i^2}$$

in view of Lemma 3.1.9 and (3.4.8), and since $G_iu_i=-\nabla f(x_i)$ while $J_iu_i=\nabla f(x_{i+1})-\nabla f(x_i)$. Hence, we can write

$$\prod_{i=0}^{k-1}\nu_i^2\ge\frac{g_k^2}{g_0^2}\prod_{i=0}^{k-1}(1+\xi_{i+1})^{-1}\overset{(3.4.6)}{\ge}(1+\xi_k)^{-k}\frac{g_k^2}{g_0^2}.$$

Consequently, according to (3.4.18),

$$\frac{13}{6}\frac{n}{k}\ln(\xi_{k+1}^{\xi_{k+1}}\varkappa)\ge\ln\Bigl(1+[(1+\xi_k)\varkappa_k]^{-1}\Bigl[\frac{g_k}{g_0}\Bigr]^{2/k}\Bigr),$$

where $\varkappa_k:=\prod_{i=0}^{k-1}p_i^{-1/k}$. Rearranging, we obtain

$$g_k\le\bigl[(1+\xi_k)\varkappa_k\bigl((\xi_{k+1}^{\xi_{k+1}}\varkappa)^{13n/(6k)}-1\bigr)\bigr]^{k/2}g_0.$$

But $\lambda_k \le \sqrt{\xi_k \varkappa}\, g_k$ by (3.4.7), and $g_0 \le \lambda_0$ in view of (3.4.2) and the fact that $G_0 = LB$. □

In the quadratic case ($M = 0$), we have $\xi_k \equiv 1$ (see (3.4.6)), and Lemmas 3.4.4 and 3.4.6 reduce to the already known Theorem 3.2.2, and Lemma 3.4.7 reduces to the already known Theorem 3.2.3. In the general case, the quantities ξ_k can grow with iterations. However, as we will see in a moment, by requiring the initial point x_0 in Algorithm 3.4.1 to be sufficiently close to the solution, we can still ensure that ξ_k stay *uniformly bounded* by a sufficiently small absolute constant. This allows us to recover all the main results of the quadratic case.

To write down the region of local convergence of Algorithm 3.4.1, we need to introduce one more quantity, related to the starting moment of superlinear convergence[8]:

$$K_0 := \left\lceil (\tau \tfrac{4}{9} \varkappa^{-1} + 1 - \tau)^{-1} 8n \ln(2\varkappa) \right\rceil, \qquad \tau := \sup_{k \ge 0} \tau_k \ (\le 1). \tag{3.4.19}$$

For DFP ($\tau_k \equiv 1$) and BFGS ($\tau_k \equiv 0$), we have respectively

$$K_0^{\mathrm{DFP}} = \lceil 18 n \varkappa \ln(2\varkappa) \rceil, \qquad K_0^{\mathrm{BFGS}} = \lceil 8n \ln(2\varkappa) \rceil. \tag{3.4.20}$$

Now we are ready to prove the main result of this section.

Theorem 3.4.8. *Suppose that, in Algorithm 3.4.1, we have*[9]

$$M\lambda_0 \le \frac{\ln(3/2)}{(3/2)^{3/2}} \max\{(2\varkappa)^{-1}, (K_0 + 9)^{-1}\}. \tag{3.4.21}$$

Then, for all $k \ge 0$,

$$\tfrac{2}{3} \nabla^2 f(x_k) \preceq G_k \preceq \tfrac{3}{2} \varkappa \nabla^2 f(x_k), \tag{3.4.22}$$

$$\lambda_k \le \left(1 - (2\varkappa)^{-1}\right)^k \sqrt{\tfrac{3}{2}}\, \lambda_0, \tag{3.4.23}$$

and, for all $k \ge 1$,

$$\lambda_k \le \left[\tfrac{5}{2} \varkappa_k \left((2\varkappa)^{13n/(6k)} - 1 \right) \right]^{k/2} \sqrt{\tfrac{3}{2} \varkappa}\, \lambda_0, \tag{3.4.24}$$

[8] Hereinafter, $\lceil t \rceil$ for $t > 0$ denotes the smallest positive integer greater or equal to t.

[9] Recall that M is the parameter of strong self-concordance defined in (3.4.3). In particular, according to Lemma 3.3.2, $M \le L_2/\mu^{3/2}$, where μ and L_2 are, respectively, the strong convexity parameter and the Lipschitz constant of the Hessian of f.

where $\varkappa_k := \prod_{i=0}^{k-1} (\tau_i \frac{4}{9} \varkappa^{-1} + 1 - \tau_i)^{-1/k}$.

Proof. Let us prove by induction that, for all $k \geq 0$, we have

$$\xi_k \leq \frac{3}{2}. \tag{3.4.25}$$

Clearly, (3.4.25) is satisfied for $k = 0$ since $\xi_0 = 1$. It is also satisfied for $k = 1$ since

$$\xi_1 \overset{(3.4.6)}{=} \exp(Mr_0) \overset{(3.4.10)}{\leq} \exp(\xi_0 M \lambda_0) \overset{(3.4.6)}{=} \exp(M\lambda_0) \overset{(3.4.21)}{\leq} \frac{3}{2}.$$

Now let $k \geq 0$, and suppose that (3.4.25) has already been proved for all indices up to $k+1$. Then, applying Lemma 3.4.4, we obtain (3.4.22) for all indices up to $k+1$. Applying now Lemma 3.4.6 and using for all $0 \leq i \leq k$ the relation

$$\begin{aligned} q_i &\overset{(3.4.12)}{=} \max\{1 - (\xi_{i+1}\varkappa)^{-1}, \xi_{i+1} - 1\} \\ &\overset{(3.4.25)}{\leq} \max\{1 - (\tfrac{3}{2}\varkappa)^{-1}, \tfrac{1}{2}\} \leq 1 - (2\varkappa)^{-1}, \end{aligned}$$

we obtain (3.4.23) for all indices up to $k+1$. Finally, if $k \geq 1$, then, applying Lemma 3.4.7 and using that, according to (3.4.25),

$$\xi_{i+1}^{\xi_{i+1}} \leq \Big(\frac{3}{2}\Big)^{3/2} = \frac{3}{2}\sqrt{\frac{3}{2}} \leq \frac{3}{2}\Big(1 + \frac{1}{4}\Big) = \frac{15}{8} \leq 2$$

for all $0 \leq i \leq k$, we obtain (3.4.24) for all indices up to k. Thus, at this moment, (3.4.22) and (3.4.23) are proved for all indices up to $k+1$, while (3.4.24) is proved only up to k.

To finish the inductive step, it remains to prove that (3.4.25) is satisfied for the index $k+2$, or, equivalently, in view of (3.4.6), that

$$M \sum_{i=0}^{k+1} r_i \leq \ln \frac{3}{2}.$$

Since

$$M \sum_{i=0}^{k+1} r_i \leq M \sum_{i=0}^{k+1} \xi_i \lambda_i \leq \frac{3}{2} M \sum_{i=0}^{k+1} \lambda_i$$

in view of (3.4.10) and (3.4.25), it suffices to show that

$$\frac{3}{2}M\sum_{i=0}^{k+1}\lambda_i \le \ln\frac{3}{2}.$$

In view of (3.4.23), we have

$$\frac{3}{2}M\sum_{i=0}^{k+1}\lambda_i \le \Big(\frac{3}{2}\Big)^{3/2}M\lambda_0\sum_{i=0}^{k+1}\big(1-(2\varkappa)^{-1}\big)^i \le \Big(\frac{3}{2}\Big)^{3/2}2\varkappa M\lambda_0. \quad (3.4.26)$$

Therefore, if we could prove that

$$\frac{3}{2}M\sum_{i=0}^{k+1}\lambda_i \le \Big(\frac{3}{2}\Big)^{3/2}(K_0+9)M\lambda_0, \quad (3.4.27)$$

then, combining (3.4.26) and (3.4.27), we would obtain

$$\frac{3}{2}M\sum_{i=0}^{k+1}\lambda_i \le \Big(\frac{3}{2}\Big)^{3/2}\min\{2\varkappa, K_0+9\}M\lambda_0 \overset{(3.4.21)}{\le} \ln\frac{3}{2},$$

which is exactly what we need. Let us prove (3.4.27). If $k \le K_0$, then, in view of (3.4.23), we have

$$\frac{3}{2}M\sum_{i=0}^{k+1}\lambda_i \le \Big(\frac{3}{2}\Big)^{3/2}(k+2)M\lambda_0 \le \Big(\frac{3}{2}\Big)^{3/2}(K_0+2)M\lambda_0,$$

and (3.4.27) follows. Therefore, from now on, we can assume that $k \ge K_0$. Then, using (3.4.23), we obtain[10]

$$\begin{aligned}\frac{3}{2}M\sum_{i=0}^{k+1}\lambda_i &= \frac{3}{2}M\Big(\sum_{i=0}^{K_0-1}\lambda_i+\lambda_{k+1}\Big)+\frac{3}{2}M\sum_{i=K_0}^{k}\lambda_i\\ &\le \Big(\frac{3}{2}\Big)^{3/2}(K_0+1)M\lambda_0+\frac{3}{2}M\sum_{i=K_0}^{k}\lambda_i.\end{aligned}$$

[10] We will estimate the second sum using (3.4.24). However, recall that, at this moment, (3.4.24) is proved only up to the index k. This is the reason why we move λ_{k+1} into the first sum.

It remains to show that

$$\frac{3}{2}M\sum_{i=K_0}^{k}\lambda_i \le \Big(\frac{3}{2}\Big)^{3/2} 8M\lambda_0.$$

We can do this using (3.4.24).

First, let us make some estimations. Clearly, for all $0 < t < 1$, we have

$$\exp(t) = \sum_{j=0}^{\infty}\frac{t^j}{j!} \le 1 + t + \frac{t^2}{2}\sum_{j=0}^{\infty} t^j = 1 + t\Big(1 + \frac{t}{2(1-t)}\Big).$$

Hence, for all $0 < t \le 1$, we obtain

$$\exp\Big(\frac{13t}{48}\Big) - 1 \le \frac{13t}{48}\Big(1 + \frac{13/48}{2(1-13/48)}\Big) = \frac{13t}{48}\cdot\frac{83}{70} \le \frac{13t}{48}\cdot\frac{6}{5} = \frac{13t}{40},$$

and so

$$\Big[\frac{5}{2t}\Big(\exp\Big\{\frac{13t}{48}\Big\} - 1\Big)\Big]^{1/2} \le \sqrt{\frac{5}{2t}\cdot\frac{13t}{40}} = \sqrt{\frac{13}{16}} \le \frac{11}{12}. \tag{3.4.28}$$

Further, since $K_0 \ge 8\ln(2\varkappa)$ in view of (3.4.19), we have

$$\begin{aligned}\Big(\frac{11}{12}\Big)^{K_0}\sqrt{\varkappa} &= \exp\Big\{K_0\ln\frac{11}{12}\Big\}\sqrt{\varkappa} \le \exp\Big\{-\frac{1}{12}K_0\Big\}\sqrt{\varkappa} \\ &\le \exp\Big\{-\frac{2}{3}\ln(2\varkappa)\Big\}\sqrt{\varkappa} = 2^{-2/3}\varkappa^{-1/6} \le 2^{-2/3} \le \frac{2}{3}.\end{aligned} \tag{3.4.29}$$

Thus, for all $K_0 \le i \le k$, and $p := \tau\frac{4}{9}\varkappa^{-1} + 1 - \tau \le \prod_{j=0}^{i-1}(\tau_i\frac{4}{9}\varkappa^{-1} + 1 - \tau_i)^{1/i}$ (see (3.4.19)), we have

$$\begin{aligned}\lambda_i &\overset{(3.4.24)}{\le} \big[\tfrac{5}{2}p^{-1}\big(\exp\{\tfrac{13}{6}i^{-1}n\ln(2\varkappa)\} - 1\big)\big]^{i/2}\sqrt{\tfrac{3}{2}\varkappa}\,\lambda_0 \\ &\overset{(3.4.19)}{\le} \big[\tfrac{5}{2}p^{-1}\big(\exp\{\tfrac{13}{48}p\} - 1\big)\big]^{i/2}\sqrt{\tfrac{3}{2}\varkappa}\,\lambda_0 \overset{(3.4.28)}{\le} (\tfrac{11}{12})^i\sqrt{\tfrac{3}{2}\varkappa}\,\lambda_0 \\ &= (\tfrac{11}{12})^{i-K_0}(\tfrac{11}{12})^{K_0}\sqrt{\tfrac{3}{2}\varkappa}\,\lambda_0 \overset{(3.4.29)}{\le} (\tfrac{11}{12})^{i-K_0}\tfrac{2}{3}\sqrt{\tfrac{3}{2}}\,\lambda_0.\end{aligned}$$

Hence,

$$\frac{3}{2}M\sum_{i=K_0}^{k}\lambda_i \le \Big(\frac{3}{2}\Big)^{3/2}M\lambda_0\cdot\frac{2}{3}\sum_{i=K_0}^{k}\Big(\frac{11}{12}\Big)^{i-K_0} \le \Big(\frac{3}{2}\Big)^{3/2}8M\lambda_0. \qquad \square$$

Comparing Theorem 3.4.8 with Theorems 3.2.2 and 3.2.3, we see that, in the general nonlinear case, we have obtained exactly the same efficiency estimates as in the quadratic case, up to some absolute constants. The only principal difference between these two cases is that, in the nonlinear case, we have to additionally require that the initial point x_0 is sufficiently good in the sense that (3.4.21) holds.

Interestingly, the region of local convergence, specified by (3.4.21), depends on the *maximum* of two quantities: $\varkappa^{-1}$ and K_0^{-1}. For DFP, the K_0^{-1} part in this maximum is, in fact, redundant, and the size of the corresponding region of local convergence is simply inversely proportional to the condition number:

$$M\lambda_0 \leq O(\varkappa^{-1}).$$

However, for BFGS, the K_0^{-1} part does not disappear, and we obtain the following region of local convergence:

$$M\lambda_0 \leq O(1)\max\{\varkappa^{-1}, [n\ln(2\varkappa)]^{-1}\}.$$

Clearly, the latter region can be much bigger than the former when the condition number $\varkappa$ is significantly larger than the dimension n.

Example 3.4.9. Consider the functions

$$f(x) \coloneqq f_0(x) + \frac{\mu}{2}\|x\|^2, \quad f_0(x) \coloneqq \ln\Big(\sum_{i=1}^m \exp(\langle a_i, x\rangle + b_i)\Big), \quad x \in \mathbb{E},$$

where $a_i \in \mathbb{E}^*$, $b_i \in \mathbb{R}$, $i = 1, \ldots, m$, $\mu > 0$, and $\|\cdot\|$ is a Euclidean norm. Let $\gamma > 0$ be such that

$$\|a_i\|_* \leq \gamma, \qquad i = 1, \ldots, m.$$

Define

$$\pi_i(x) \coloneqq \frac{\exp(\langle a_i, x\rangle + b_i)}{\sum_{j=1}^m \exp(\langle a_j, x\rangle + b_j)}, \qquad x \in \mathbb{E}, \quad i = 1, \ldots, m.$$

Clearly, $\sum_{i=1}^m \pi_i(x) = 1$, $\pi_i(x) > 0$ for all $x \in \mathbb{E}$, $i = 1, \ldots, m$. It is not difficult to check that, for all $x, h \in \mathbb{E}$, we have

$$\langle \nabla f_0(x), h\rangle = \sum_{i=1}^m \pi_i(x)\langle a_i, h\rangle \leq \gamma,$$

$$\begin{aligned}\langle \nabla^2 f_0(x)h, h\rangle &= \sum_{i=1}^m \pi_i(x)\langle a_i - \nabla f_0(x), h\rangle^2 \\ &= \sum_{i=1}^m \pi_i(x)\langle a_i, h\rangle^2 - \langle \nabla f_0(x), h\rangle^2 \le \gamma^2\|h\|^2, \\ D^3 f_0(x)[h]^3 &= \sum_{i=1}^m \pi_i(x)\langle a_i - \nabla f_0(x), h\rangle^3 \\ &\le 2\gamma\|h\|\langle \nabla^2 f_0(x)h, h\rangle \le 2\gamma^3\|h\|^3.\end{aligned}$$

Thus, f_0 is a convex function with γ^2-Lipschitz gradient and $(2\gamma^3)$-Lipschitz Hessian. Consequently, the function f is μ-strongly convex with L-Lipschitz gradient, $(2\gamma^3)$-Lipschitz Hessian, and, in view of Lemma 3.3.2, M-strongly self-concordant, where

$$L := \gamma^2 + \mu, \qquad M := 2\gamma^3\mu^{-3/2}.$$

Let the regularization parameter μ be sufficiently small, namely, such that

$$\bar{\varkappa} := \frac{\gamma^2}{\mu} \ge 1.$$

Then,

$$\bar{\varkappa} \le \varkappa \le 2\bar{\varkappa}, \qquad M = 2\bar{\varkappa}^{3/2},$$

where $\varkappa := L/\mu$. Hence, according to (3.4.21), the region of local convergence of BFGS can be described as follows:

$$\lambda_0 \le O(1)\max\bigl\{\bar{\varkappa}^{-5/2}, [n\bar{\varkappa}^{3/2}\ln(4\bar{\varkappa})]^{-1}\bigr\}.$$

3.5 Discussion

We have obtained explicit rates of local superlinear convergence for the classical quasi-Newton methods from the convex Broyden class. The main parameters in these rates are the dimension n of the problem and its condition number $\varkappa$.

For the important BFGS and DFP methods, the principal factors in the corresponding complexity estimates are as follows (up to some absolute constants):

$$\begin{aligned}&\text{BFGS:} \quad n\ln\varkappa, \\ &\text{DFP:} \quad n\varkappa\ln\varkappa.\end{aligned} \tag{3.5.1}$$

According to these estimates, BFGS is almost insensitive to the condition number, and its efficiency mainly depends on the dimension of the problem. In contrast, for the DFP Method, the condition number is outside the logarithm, which means that DFP may have poor performance on ill-conditioned problems. This theoretical conclusion confirms the well-known empirical superiority of the BFGS Method over DFP.

Note that the results, presented in this chapter, are *local*, i.e., they are valid under the assumption that the starting point is sufficiently close to the minimizer. In particular, there is no contradiction between these results and the fact that DFP is not known to be globally convergent with inexact line search (see, e.g., [25]).

To conclude, let us mention some open questions. First, looking at the complexity estimate (3.5.1) for the BFGS Method, in addition to the dimension of the problem, we see the presence of the logarithm of its condition number. Although typically such logarithmic factors are considered small, it is still interesting to understand whether this factor can be completely removed.

Second, in all the methods we have considered, the initial Hessian approximation G_0 was LB, where L is the Lipschitz constant of the gradient, measured w.r.t. the operator B. We always assume that this constant is known. Of course, it is interesting to develop some *adaptive* algorithms (similar to Algorithm 2.3.2), which could start from any initial guess L_0 for the constant L, and then somehow dynamically adjust the Hessian approximations in iterations, yet retaining all the original efficiency estimates.

Finally, it is also interesting whether the results obtained in this chapter can be applied to *limited-memory* quasi-Newton methods such as L-BFGS [113]. Unfortunately, it seems that the answer is negative. The main problem is that we cannot say anything interesting about just a *few* iterations of, say, the standard BFGS. Indeed, according to our main result, after k iterations of the method, the initial residual is contracted by the factor of the form $[\varkappa^{n/k} - 1]^k$. For all values of $k \leq n \ln \varkappa$, this contraction factor is, in fact, bigger than 1, so the estimate is practically useless.

3.A Appendix

3.A.1 Proof of Lemma 3.1.1

i. Denote $\varphi \coloneqq \varphi_\tau(A, G, u)$ and

$$G_0 \coloneqq G - \frac{Guu^*G}{\langle Gu, u\rangle} + \frac{Auu^*A}{\langle Au, u\rangle}, \qquad s \coloneqq \frac{Au}{\langle Au, u\rangle} - \frac{Gu}{\langle Gu, u\rangle}. \tag{3.A.1}$$

According to (3.1.3), (3.1.2) and (3.1.1), we have

$$\begin{aligned} G_+ &= G_0 + \varphi\left[\frac{\langle Gu, u\rangle Auu^*A}{\langle Au, u\rangle^2} + \frac{Guu^*G}{\langle Gu, u\rangle} - \frac{Auu^*G + Guu^*A}{\langle Au, u\rangle}\right] \\ &= G_0 + \varphi\langle Gu, u\rangle ss^*. \end{aligned} \tag{3.A.2}$$

ii. Let us compute $\det(G^{-1}, G_0)$. Let $M \coloneqq G + \frac{Auu^*A}{\langle Au,u\rangle}$. Note that

$$Mu = Gu + Au, \qquad MG^{-1}Au = \left(\frac{\langle AG^{-1}Au, u\rangle}{\langle Au, u\rangle} + 1\right)Au, \tag{3.A.3}$$

and $G_0 = M - \frac{Guu^*G}{\langle Gu,u\rangle}$. Applying Proposition 2.1.5 twice, we obtain

$$\begin{aligned} \det(G^{-1}, M) &= 1 + \frac{\langle AG^{-1}Au, u\rangle}{\langle Au, u\rangle}, \\ \det(M^{-1}, G_0) &= 1 - \frac{\langle GM^{-1}Gu, u\rangle}{\langle Gu, u\rangle} \overset{(3.A.3)}{=} 1 - \frac{\langle Gu - GM^{-1}Au, u\rangle}{\langle Gu, u\rangle} \\ &= \frac{\langle GM^{-1}Au, u\rangle}{\langle Gu, u\rangle} \overset{(3.A.3)}{=} \left(\frac{\langle AG^{-1}Au, u\rangle}{\langle Au, u\rangle} + 1\right)^{-1} \frac{\langle Au, u\rangle}{\langle Gu, u\rangle}. \end{aligned}$$

Thus,

$$\det(G^{-1}, G_0) = \det(G^{-1}, M)\det(M^{-1}, G_0) = \frac{\langle Au, u\rangle}{\langle Gu, u\rangle}. \tag{3.A.4}$$

iii. Let us show that

$$G_0^{-1} = \Big(I_{\mathbb{E}} - \frac{uu^*A}{\langle Au, u\rangle}\Big)G^{-1}\Big(I_{\mathbb{E}^*} - \frac{Auu^*}{\langle Au, u\rangle}\Big) + \frac{uu^*}{\langle Au, u\rangle}, \tag{3.A.5}$$

where $I_{\mathbb{E}}$ and $I_{\mathbb{E}^*}$ are the identity operators in $\mathbb{E}$ and $\mathbb{E}^*$, respectively. Indeed, denote the right-hand side in (3.A.5) by H_0. Using that $G_0u = Au$,

we obtain

$$\begin{aligned} H_0G_0 &= \Big(I_{\mathbb{E}} - \frac{uu^*A}{\langle Au,u\rangle}\Big)G^{-1}\Big(G_0 - \frac{Auu^*A}{\langle Au,u\rangle}\Big) + \frac{uu^*A}{\langle Au,u\rangle} \\ &= \Big(I_{\mathbb{E}} - \frac{uu^*A}{\langle Au,u\rangle}\Big)G^{-1}\Big(G - \frac{Guu^*G}{\langle Gu,u\rangle}\Big) + \frac{uu^*A}{\langle Au,u\rangle} = I_{\mathbb{E}}. \end{aligned}$$

iv. From (3.A.1), it follows that

$$\langle s,u\rangle = 0, \tag{3.A.6}$$

$$\langle Au, G^{-1}s\rangle = \frac{\langle AG^{-1}Au,u\rangle}{\langle Au,u\rangle} - \frac{\langle Au,u\rangle}{\langle Gu,u\rangle}. \tag{3.A.7}$$

Combining (3.A.5)–(3.A.7) and (3.A.1), we obtain

$$\begin{aligned} \langle Au,u\rangle G_0^{-1}s &= \langle Au,u\rangle G^{-1}s - \langle Au, G^{-1}s\rangle u \\ &= \langle Au,u\rangle G^{-1}s - \left(\frac{\langle AG^{-1}Au,u\rangle}{\langle Au,u\rangle} - \frac{\langle Au,u\rangle}{\langle Gu,u\rangle}\right)u \\ &= G^{-1}Au - \frac{\langle AG^{-1}Au,u\rangle}{\langle Au,u\rangle}u, \end{aligned} \tag{3.A.8}$$

Consequently, in view of (3.A.6) and (3.A.7),

$$\langle Au,u\rangle\langle s, G_0^{-1}s\rangle = \langle Au, G^{-1}s\rangle = \frac{\langle AG^{-1}Au,u\rangle}{\langle Au,u\rangle} - \frac{\langle Au,u\rangle}{\langle Gu,u\rangle}. \tag{3.A.9}$$

Applying now Proposition 2.1.5 to (3.A.2) and using (3.A.9), we get

$$\begin{aligned} \det(G_0^{-1}, G_+) &= 1 + \varphi\langle Gu,u\rangle\langle s, G_0^{-1}s\rangle \\ &= 1 + \varphi\frac{\langle Gu,u\rangle}{\langle Au,u\rangle}\left(\frac{\langle AG^{-1}Au,u\rangle}{\langle Au,u\rangle} - \frac{\langle Au,u\rangle}{\langle Gu,u\rangle}\right) \\ &= \frac{\langle Gu,u\rangle}{\langle Au,u\rangle}\left[\varphi\frac{\langle AG^{-1}Au,u\rangle}{\langle Au,u\rangle} + (1-\varphi)\frac{\langle Au,u\rangle}{\langle Gu,u\rangle}\right] \\ &= \left[\tau\frac{\langle Au,u\rangle}{\langle AG^{-1}Au,u\rangle} + (1-\tau)\frac{\langle Gu,u\rangle}{\langle Au,u\rangle}\right]^{-1}, \end{aligned} \tag{3.A.10}$$

where the last identity follows from the definition of φ in (3.1.4).

Combining (3.A.4) and (3.A.10), we conclude that

$$\det(G^{-1}, G_+) = \det(G^{-1}, G_0)\det(G_0^{-1}, G_+)$$

$$= \left[\tau \frac{\langle Au, u\rangle}{\langle AG^{-1}Au, u\rangle} + (1-\tau)\frac{\langle Gu, u\rangle}{\langle Au, u\rangle}\right]^{-1}.$$

This proves (3.1.6).

v. Applying Proposition 2.1.1 to (3.A.2), we obtain

$$G_+^{-1} = G_0^{-1} - \frac{\varphi\langle Gu, u\rangle}{1 + \varphi\langle Gu, u\rangle\langle s, G_0^{-1}s\rangle} G_0^{-1}ss^*G_0^{-1}.$$

From (3.A.10) and (3.1.4), we see that

$$\begin{aligned}
&\frac{\varphi\langle Gu, u\rangle}{1 + \varphi\langle Gu, u\rangle\langle s, G_0^{-1}s\rangle} \\
&\quad = \varphi\langle Gu, u\rangle \frac{\langle Au, u\rangle}{\langle Gu, u\rangle}\left[\tau \frac{\langle Au, u\rangle}{\langle AG^{-1}Au, u\rangle} + (1-\tau)\frac{\langle Gu, u\rangle}{\langle Au, u\rangle}\right] \\
&\quad = \tau \frac{\langle Au, u\rangle^2}{\langle AG^{-1}Au, u\rangle}.
\end{aligned}$$

Thus,

$$G_+^{-1} = G_0^{-1} - \tau \frac{\langle Au, u\rangle^2}{\langle AG^{-1}Au, u\rangle} G_0^{-1}ss^*G_0^{-1}.$$

Substituting (3.A.8), we get

$$\begin{aligned}
G_+^{-1} = G_0^{-1} - \tau\Bigg[&\frac{G^{-1}Auu^*AG^{-1}}{\langle AG^{-1}Au, u\rangle} - \frac{G^{-1}Auu^* + uu^*AG^{-1}}{\langle Au, u\rangle} \\
&+ \frac{\langle AG^{-1}Au, u\rangle uu^*}{\langle Au, u\rangle^2}\Bigg].
\end{aligned}$$

Using now (3.A.5) and grouping the terms, we obtain (3.1.5). □

3.A.2 Auxiliary Operator Inequality

Lemma 3.A.1. *Let $A, B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ be such that*

$$A \preceq B. \tag{3.A.11}$$

Then, for any $u \in \mathbb{E} \setminus \{0\}$, we have

$$A - \frac{Auu^*A}{\langle Au, u\rangle} \preceq B - \frac{Buu^*B}{\langle Bu, u\rangle}.$$

Proof. Indeed, for all $h \in \mathbb{E}$, we have

$$\begin{aligned}
\langle Ah, h\rangle - \frac{\langle Au, h\rangle^2}{\langle Au, u\rangle} &= \min_{\alpha\in\mathbb{R}} \left[\langle Ah, h\rangle - 2\alpha\langle Ah, u\rangle + \alpha^2\langle Au, u\rangle\right] \\
&= \min_{\alpha\in\mathbb{R}} \langle A(h - \alpha u), h - \alpha u\rangle \\
&\leq \min_{\alpha\in\mathbb{R}} \langle B(h - \alpha u), h - \alpha u\rangle \\
&= \min_{\alpha\in\mathbb{R}} \left[\langle Bh, h\rangle - 2\alpha\langle Bh, u\rangle + \alpha^2\langle Bu, u\rangle\right] \\
&= \langle Bh, h\rangle - \frac{\langle Bu, h\rangle^2}{\langle Bu, u\rangle},
\end{aligned}$$

where the inequality follows from (3.A.11). □

Chapter 4

Greedy Quasi-Newton Methods

In Chapter 3, we have studied the local convergence of classical quasi-Newton methods for smooth optimization. The main property responsible for their superlinear convergence is that the Hessian approximations produced by the methods converge to the true Hessians *along search directions*. However, in some situations[1], it is desirable to have the convergence of Hessian approximations to the exact Hessians in the traditional sense, i.e., along *any* direction. Unfortunately, classical quasi-Newton methods, in general[2], are not able to ensure such convergence (see, e.g., [45]).

In this chapter, we propose new *greedy* quasi-Newton methods which are free of this drawback. Specifically, they generate Hessian approximations whose deviation from the exact Hessians converges to zero at a linear rate. Furthermore, the rate of superlinear convergence of greedy quasi-Newton methods is asymptotically faster than that of the classical ones.

The main difference between greedy and classical quasi-Newton methods is the choice of the direction in the update formula for Hessian approximation. In classical methods, this direction is chosen as the difference of successive iterates, while, in greedy methods, this is instead chosen as a certain *basis vector*, greedily selected to optimize some measure of progress.

It is worth mentioning that the idea of using basis vectors in quasi-

[1] One example could be the application of quasi-Newton methods for solving auxiliary subproblems arising in path-following interior-point methods.

[2] However, it is worth mentioning that there are some settings in which the standard SR1 Method indeed yields convergence to the true Hessian (for more details, see [34]).

Newton methods for approximating the Hessian goes back at least to so-called *methods of dual directions*[3] (see [153]). For these methods, it is also possible to prove both local superlinear convergence of the iterates and convergence of Hessian approximations. However, as in standard quasi-Newton methods, the corresponding results are only asymptotic. Nevertheless, despite the fact that the greedy quasi-Newton methods, presented in this chapter, are based on a similar idea, their construction and analysis are significantly different. In particular, methods of dual directions do not use updating formulas from the Broyden class, and work with Hessian approximations that may not be self-adjoint.

One should also mention that some randomized variants of quasi-Newton algorithms have been proposed recently, which also use nonstandard directions for updating Hessian approximations [73, 74, 100].

Contents

This chapter is organized as follows. In Section 4.1, we discuss a class of quasi-Newton updating rules for approximating a self-adjoint positive definite linear operator. We present a special greedy strategy for selecting an update direction, which ensures a linear convergence rate in approximating the target operator. In Section 4.2, we analyze greedy quasi-Newton methods, applied to the problem of minimizing a quadratic function. We show that these methods have a global linear convergence rate, comparable to that of standard gradient descent, and also a superlinear convergence rate, which contains a contraction factor depending on the square of the iteration counter. In Section 4.3, we show that similar results also hold in a more general setting of minimizing a strongly convex and strongly self-concordant function with Lipschitz gradient, provided that the starting point is chosen sufficiently close to the solution. The main difficulty here, compared to the quadratic case, is that the Hessian of the objective function is no longer constant, resulting in the need to apply a special *correction strategy* to keep Hessian approximations under control. In Section 4.4, we compare the effi-

[3] One particular method of this type suggests approximating the Hessian $\nabla^2 f(x_k)$ at each iteration k with an operator G_k whose action on each basis vector e_j $(1 \le j \le n)$ approximates that of $\nabla^2 f(\hat{x}_{k,j})$ on e_j, where $\hat{x}_{k,j} \in \{x_{k-n+1}, \dots, x_k\}$ is one of the previous n points. More specifically, G_k is chosen as the solution of the following system of linear equations: $G_k r_{k-i} = \delta_{k-i}$, where $r_t := e_{(t \bmod n)+1}$ for any $t \ge 0$ is the cyclic repetition of basis vectors, and $\delta_t := [\nabla f(x_t + h_t r_t) - \nabla f(x_t)]/h_t$ for any $t \ge 0$ is a finite difference approximation of $\nabla^2 f(x_t) r_t$ with a certain "discretization step" $h_t > 0$ (such that $h_t \to 0$ as $t \to \infty$). In the end, G_{k+1} differs from G_k by a rank-one operator, and thus G_{k+1}^{-1} can be efficiently computed from G_k^{-1}.

ciency estimates we have for the greedy quasi-Newton methods with those of the classical methods. Finally, in Section 4.5, we present some preliminary computational results.

The contents of this chapter is based on [159], with the following minor modifications. First, we have introduced a new name "extended convex Broyden class" (and slightly different notation) for the special subclass of the Broyden family, defined in Section 4.1 and used throughout this chapter. Second, we have removed the definition of strongly self-concordant functions and the discussion of their properties, as this material was already presented in Section 3.3. Third, we have included a new Section 4.4 with the comparison of the efficiency estimates of greedy quasi-Newton methods with those of the classical ones.

4.1 Greedy Quasi-Newton Updates

In this chapter, we will be working with a certain subclass of the Broyden family, which is bigger than the standard convex Broyden class, and which, in particular, includes all three most famous quasi-Newton updates: SR1, BFGS and DFP. However, in order to handle such a large class properly, we will need to make a certain extra assumption on the relation between the target operator and its quasi-Newton approximation.

Let us present the main definitions. As in Chapter 3, in this chapter, we work with the operator-revealing form of quasi-Newton updates. Let $A \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ be the target operator which we want to approximate, and let $G \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ be its current approximation. Our main extra assumption is that G is an *upper* approximation of A:

$$A \preceq G. \tag{4.1.1}$$

In what follows, we always assume that (4.1.1) is satisfied.

Consider the following class of quasi-Newton updates of G w.r.t. A along a direction $u \in \mathbb{E} \setminus \ker(G - A)$, parametrized by a scalar $\chi \in \mathbb{R}$:

$$\mathrm{EBroyd}_\chi(A, G, u) := (1 - \chi)\,\mathrm{SR1}(A, G, u) + \chi\,\mathrm{DFP}(A, G, u), \tag{4.1.2}$$

where $\mathrm{SR1}(A, G, u)$ and $\mathrm{DFP}(A, G, u)$ are, respectively, the SR1 and DFP

updates of G w.r.t. A along u:

$$\mathrm{SR1}(A, G, u) \coloneqq G - \frac{(G-A)uu^*(G-A)}{\langle (G-A)u, u\rangle}, \tag{4.1.3}$$

$$\mathrm{DFP}(A, G, u) \coloneqq G - \frac{Auu^*G + Guu^*A}{\langle Au, u\rangle} + \left(\frac{\langle Gu, u\rangle}{\langle Au, u\rangle} + 1\right)\frac{Auu^*A}{\langle Au, u\rangle}. \tag{4.1.4}$$

Note that, under our main assumption (4.1.1), for any $u \in \mathbb{E} \setminus \ker(G-A)$, the denominators in (4.1.3) and (4.1.4) are nonzero and therefore all the updates in (4.1.2)–(4.1.4) are well-defined. For the sake of convenience, we also set $\mathrm{EBroyd}_\chi(A, G, u) \coloneqq G$ for any $u \in \ker(G-A)$.

By definition, the family (4.1.2) is a line passing through the SR1 and DFP updates. However, as we know from Section 2.5.2, this line is actually the Broyden class. Thus, (4.1.2) is an alternative parametrization of the usual Broyden class.

In this chapter, our interest will be in the subclass of (4.1.2) described by the values of $\chi \in [0, 1]$. In what follows, we will refer to this subclass as the *extended convex Broyden class*. Let us emphasize once again that we consider this class exclusively under the extra assumption (4.1.1).

Geometrically, the extended convex Broyden class is a segment between the SR1 and DFP updates. The name comes from the fact that this segment also contains the BFGS update, and hence the whole convex Broyden class. Indeed, for

$$\chi_{\mathrm{BFGS}} \coloneqq \frac{\langle Au, u\rangle}{\langle Gu, u\rangle} \overset{(4.1.1)}{\in} (0, 1), \tag{4.1.5}$$

we have, according to (4.1.2)–(4.1.4),

$$\begin{aligned}
&\mathrm{EBroyd}_{\chi_{\mathrm{BFGS}}}(A, G, u)\\
&\quad= G - \frac{\langle (G-A)u, u\rangle}{\langle Gu, u\rangle}\frac{(G-A)uu^*(G-A)}{\langle (G-A)u, u\rangle}\\
&\qquad+ \frac{\langle Au, u\rangle}{\langle Gu, u\rangle}\left[-\frac{Auu^*G + Guu^*A}{\langle Au, u\rangle} + \left(\frac{\langle Gu, u\rangle}{\langle Au, u\rangle} + 1\right)\frac{Auu^*A}{\langle Au, u\rangle}\right]\\
&\quad= G - \frac{(G-A)uu^*(G-A)}{\langle Gu, u\rangle} - \frac{Auu^*G + Guu^*A}{\langle Gu, u\rangle}\\
&\qquad+ \left(\frac{\langle Gu, u\rangle}{\langle Au, u\rangle} + 1\right)\frac{Auu^*A}{\langle Gu, u\rangle} = G - \frac{Guu^*G}{\langle Gu, u\rangle} + \frac{Auu^*A}{\langle Au, u\rangle}.
\end{aligned}$$

This is exactly the BFGS update which we already saw in (3.1.1).

As we will see shortly, the extended convex Broyden class shares some

similar properties with the standard convex Broyden class. At the very least, each update from this class—an *extended convex Broyden update*—preserves the main assumption (4.1.1), and, in particular, positive definiteness (see Lemma 4.1.2). But first let us establish an auxiliary monotonicity result.

Lemma 4.1.1. *For any $A, G \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$, such that $A \preceq G$, any $u \in \mathbb{E}$, and any $\chi_1, \chi_2 \in \mathbb{R}$, the following implication holds:*

$$\chi_1 \leq \chi_2 \quad \Longrightarrow \quad \mathrm{EBroyd}_{\chi_1}(A, G, u) \preceq \mathrm{EBroyd}_{\chi_2}(A, G, u).$$

Proof. Suppose that $u \notin \ker(G - A)$ since otherwise the claim is trivial. According to (4.1.2)–(4.1.4), we have

$$\begin{aligned}
&\mathrm{EBroyd}_{\chi}(A, G, u) = G - \frac{(G-A)uu^*(G-A)}{\langle (G-A)u, u\rangle} \\
&+\chi\left[\frac{(G-A)uu^*(G-A)}{\langle (G-A)u, u\rangle} - \frac{Auu^*G + Guu^*A}{\langle Au, u\rangle} + \left(\frac{\langle Gu, u\rangle}{\langle Au, u\rangle} + 1\right)\frac{Auu^*A}{\langle Au, u\rangle}\right].
\end{aligned}$$

Denote

$$s \coloneqq \frac{(G-A)u}{\langle (G-A)u, u\rangle} - \frac{Au}{\langle Au, u\rangle}.$$

Then,

$$\begin{aligned}
&\langle (G-A)u, u\rangle ss^* \\
&\quad = \frac{(G-A)uu^*(G-A)}{\langle (G-A)u, u\rangle} + \frac{\langle (G-A)u, u\rangle}{\langle Au, u\rangle}\frac{Auu^*A}{\langle Au, u\rangle} \\
&\qquad\qquad - \frac{(G-A)uu^*A + Auu^*(G-A)}{\langle Au, u\rangle} \\
&\quad = \frac{(G-A)uu^*(G-A)}{\langle (G-A)u, u\rangle} - \frac{Auu^*G + Guu^*A}{\langle Au, u\rangle} + \left(\frac{\langle Gu, u\rangle}{\langle Au, u\rangle} + 1\right)\frac{Auu^*A}{\langle Au, u\rangle}.
\end{aligned}$$

Therefore,

$$\mathrm{EBroyd}_{\chi}(A, G, u) = G - \frac{(G-A)uu^*(G-A)}{\langle (G-A)u, u\rangle} + \chi\langle (G-A)u, u\rangle ss^*.$$

The claim now follows from the fact that $\langle (G-A)u, u\rangle ss^* \succeq 0$. □

Recall from Lemma 3.1.3 that each convex Broyden update preserves the bounds on the eigenvalues w.r.t. the target operator. For an extended convex Broyden update, we have the following slightly weaker variant of

this result.

Lemma 4.1.2. *Let $A, G \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ and $\eta \geq 1$ be such that*

$$A \preceq G \preceq \eta A. \tag{4.1.6}$$

Then, for any $u \in \mathbb{E}$ and any $\chi \in [0, 1]$, we have

$$A \preceq \mathrm{EBroyd}_\chi(A, G, u) \preceq \eta A. \tag{4.1.7}$$

Proof. We can assume that $u \notin \ker(G - A)$ since otherwise the claim is trivial. In view of Lemma 4.1.1 and (4.1.2), it suffices to prove independently the following two inequalities, assuming that (4.1.6) holds:

$$\mathrm{SR1}(A, G, u) \succeq A, \qquad \mathrm{DFP}(A, G, u) \preceq \eta A.$$

For the DFP update, the inequality follows from Lemma 3.1.3. For the SR1 update, we can prove it as follows. Denote $G_+ \coloneqq \mathrm{SR1}(A, G, u)$ and $R \coloneqq G - A \succeq 0$. Then, in view of (4.1.3),

$$G_+ - A = R - \frac{Ruu^*R}{\langle Ru, u\rangle} = \Big(I_{\mathbb{E}^*} - \frac{Ruu^*}{\langle Ru, u\rangle}\Big) R \Big(I_{\mathbb{E}} - \frac{uu^*R}{\langle Ru, u\rangle}\Big) \succeq 0,$$

where $I_{\mathbb{E}}$, $I_{\mathbb{E}^*}$ are the identity operators in $\mathbb{E}$ and $\mathbb{E}^*$. □

Remark 4.1.3. Results similar to Lemma 4.1.2 have been known for some time in the literature for different quasi-Newton updating formulas. For example, in [40] and [69], it was proved for the SR1 update that if $A \preceq G$ (respectively, $G \preceq A$), then $A \preceq G_+$ (respectively, $G_+ \preceq A$), where G_+ is the result of the SR1 update.

Interestingly, from Lemmas 4.1.1 and 4.1.2, it follows, under the main assumption (4.1.1), that

$$A \preceq \mathrm{SR1}(A, G, u) \preceq \mathrm{BFGS}(A, G, u) \preceq \mathrm{DFP}(A, G, u).$$

In other words, the approximation produced by SR1, is better than the one produced by BFGS, which is in turn better than the one produced by DFP.

Let us now justify the efficiency of the extended convex Broyden update in ensuring convergence $G \to A$. For this, we introduce the following measure of progress:

$$\sigma_A(G) \coloneqq \langle A^{-1}, G - A\rangle \overset{(2.1.25)}{=} \langle A^{-1}, G\rangle - n. \tag{4.1.8}$$

Thus, $\sigma_A(G)$ is the sum of the eigenvalues of the difference $G - A$, measured w.r.t. the operator A (see Proposition 2.1.3(iv)). Clearly, for G, satisfying (4.1.1), we have $\sigma_A(G) \geq 0$ with $\sigma_A(G) = 0$ if and only if $G = A$. Therefore, we need to ensure that $\sigma_A(G) \to 0$ by choosing an appropriate sequence of update directions u.

First, let us estimate the decrease in σ_A for an arbitrary direction.

Lemma 4.1.4. *Let $A, G \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ be such that $A \preceq G$. Then, for any $u \in \mathbb{E} \setminus \{0\}$, any $\chi \in [0, 1]$ and $G_+ := \mathrm{EBroyd}_\chi(A, G, u)$, we have*

$$\sigma_A(G) - \sigma_A(G_+) \geq \frac{\langle (G - A)u, u \rangle}{\langle Au, u \rangle}. \tag{4.1.9}$$

Proof. By Lemma 4.1.1 and (4.1.4), we have

$$\begin{aligned} G - G_+ &\succeq G - \mathrm{DFP}(A, G, u) \\ &= \frac{Auu^*G + Guu^*A}{\langle Au, u \rangle} - \left(\frac{\langle Gu, u \rangle}{\langle Au, u \rangle} + 1 \right) \frac{Auu^*A}{\langle Au, u \rangle}. \end{aligned}$$

Therefore, in view of (4.1.8),

$$\begin{aligned} \sigma_A(G) - \sigma_A(G_+) &= \langle A^{-1}, G - G_+ \rangle \\ &\geq 2 \frac{\langle Gu, u \rangle}{\langle Au, u \rangle} - \left(\frac{\langle Gu, u \rangle}{\langle Au, u \rangle} + 1 \right) \\ &= \frac{\langle Gu, u \rangle}{\langle Au, u \rangle} - 1 = \frac{\langle (G - A)u, u \rangle}{\langle Au, u \rangle}. \end{aligned}$$

This is exactly (4.1.9). □

According to Lemma 4.1.4, the choice of the updating direction u directly influences the bound on the decrease in the measure σ_A. Ideally, we would like to select a direction u, which maximizes the right-hand side in (4.1.9). However, this requires finding an eigenvector, corresponding to the maximal eigenvalue of G w.r.t. A, which might be computationally a difficult problem[4]. Therefore, let us consider another approach.

[4]This is a well-known problem in Linear Algebra called the *Generalized Eigenvalue Problem* (GEP). In principle, it can be solved in $O(n^3)$ operations using standard linear algebra techniques, where n is the dimension of the space (see Section 8.7 in [72]). However, this is too expensive compared to the $O(n^2)$ complexity of a typical quasi-Newton step. Alternatively, we could use some iterative methods to find an approximate solution to GEP. However, it is difficult to guarantee that $O(n^2)$ operations will be enough for such methods to obtain a sufficiently accurate solution. That is why we are not pursuing

Let us fix in the space $\mathbb{E}$ some basis:

$$e_1, \dots, e_n \in \mathbb{E}.$$

W.r.t. this basis, we can define the following greedily selected direction:

$$\bar{u}_A(G) \coloneqq \operatorname*{argmax}_{u \in \{e_1, \dots, e_n\}} \frac{\langle (G - A)u, u \rangle}{\langle Au, u \rangle} = \operatorname*{argmax}_{u \in \{e_1, \dots, e_n\}} \frac{\langle Gu, u \rangle}{\langle Au, u \rangle}. \tag{4.1.10}$$

Thus, $\bar{u}_A(G)$ is a basis vector which maximizes the right-hand side in (4.1.9). Note that for certain choices of the basis, the computation of $\bar{u}_A(G)$ might be relatively simple. For example, if $\mathbb{E} = \mathbb{R}^n$, and $e_1, \dots, e_n$ are coordinate directions, then the calculation of $\bar{u}_A(G)$ requires computing only the *diagonals* of the matrix representations of the operators G and A. The update (4.1.2), applying the rule (4.1.10), is called the *greedy quasi-Newton update*.

Let us show that the greedy quasi-Newton update decreases the measure σ_A with a *linear* rate. For this, define

$$B \coloneqq \Big(\sum_{i=1}^{n} e_i e_i^* \Big)^{-1}. \tag{4.1.11}$$

Note that $B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$.

Theorem 4.1.5. *Let $A, G \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ be such that $A \preceq G$. Further, let $\mu, L > 0$ be such that*

$$\mu B \preceq A \preceq LB. \tag{4.1.12}$$

Then, for any $\chi \in [0, 1]$, $\bar{u} \coloneqq \bar{u}_A(G)$ and $G_+ \coloneqq \mathrm{EBroyd}_\chi(A, G, \bar{u})$, we have

$$\sigma_A(G_+) \leq \Big(1 - \frac{\mu}{nL}\Big) \sigma_A(G). \tag{4.1.13}$$

this direction any further.

Proof. Denoting $R \coloneqq G - A \succeq 0$ and applying Lemma 4.1.4, we obtain

$$\begin{aligned}
&\sigma_A(G) - \sigma_A(G_+) \\
&\quad \geq \frac{\langle R\bar{u}, \bar{u}\rangle}{\langle A\bar{u}, \bar{u}\rangle} \overset{(4.1.10)}{=} \max_{1\leq i\leq n} \frac{\langle Re_i, e_i\rangle}{\langle Ae_i, e_i\rangle} \overset{(4.1.12)}{\geq} \frac{1}{L} \max_{1\leq i\leq n} \langle Re_i, e_i\rangle \\
&\quad \geq \frac{1}{nL} \sum_{i=1}^n \langle Re_i, e_i\rangle \overset{(2.1.24)}{=} \frac{1}{nL} \sum_{i=1}^n \langle e_i e_i^*, R\rangle \\
&\overset{(4.1.11)}{=} \frac{1}{nL}\langle B^{-1}, R\rangle \overset{(4.1.12)}{\geq} \frac{\mu}{nL}\langle A^{-1}, R\rangle \overset{(4.1.8)}{=} \frac{\mu}{nL}\sigma_A(G).
\end{aligned}$$

□

Remark 4.1.6. A simple modification of the above proof shows that the factor nL in (4.1.13) can be improved up to $\langle B^{-1}, A\rangle$. However, to simplify the future analysis, we prefer to work directly with constant L.

4.2 Unconstrained Quadratic Minimization

Let us demonstrate how we can apply the quasi-Newton updates described in the previous section for minimizing the quadratic function

$$f(x) \coloneqq \tfrac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle, \qquad x \in \mathbb{E}, \tag{4.2.1}$$

where $A \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ and $b \in \mathbb{E}^*$.

Let B be the operator, defined in (4.1.11), and let $\mu, L > 0$ be such that

$$\mu B \preceq A \preceq LB. \tag{4.2.2}$$

Thus, μ is the *constant of strong convexity* of f, and L is the *Lipschitz constant* of the gradient of f, both measured w.r.t. the operator B. The ratio of these two constants is the *condition number* of the function (4.2.1):

$$\varkappa \coloneqq \frac{L}{\mu} \ (\geq 1).$$

Consider the following quasi-Newton scheme.

Algorithm 4.2.1: Extended Convex Broyden Method for Quadratic Function
Initialization: Choose $x_0 \in \mathbb{E}$. Set $G_0 = LB$.
For $k \geq 0$ **iterate:** 1. Update $x_{k+1} = x_k - G_k^{-1} \nabla f(x_k)$. 2. Choose $u_k \in \mathbb{E}$ and $\chi_k \in [0, 1]$. 3. Compute $G_{k+1} = \mathrm{EBroyd}_{\chi_k}(A, G_k, u_k)$.

Since Algorithm 4.2.1 starts with $G_0 = LB$, from (4.2.2), it follows that $A \preceq G_0$. Hence, in view of Lemma 4.1.2, we have

$$A \preceq G_k \tag{4.2.3}$$

for all $k \geq 0$. In particular, all G_k are positive definite, and Algorithm 4.2.1 is well-defined.

Remark 4.2.1. For avoiding the $O(n^3)$ operations for computing $G_k^{-1} \nabla f(x_k)$ at each iteration of Algorithm 4.2.1, when implementing this method, one should maintain the inverse Hessian approximations $H_k := G_k^{-1}$. Due to a low-rank structure of the Broyden update, H_k can be efficiently updated into H_{k+1} at the cost of $O(n^2)$.

To estimate the convergence rate of Algorithm 4.2.1, let us look at the norm of the gradient of f, measured w.r.t. A:

$$\lambda_f(x) := \|\nabla f(x)\|_A^* = \langle \nabla f(x), A^{-1} \nabla f(x) \rangle^{1/2}, \qquad x \in \mathbb{E}. \tag{4.2.4}$$

The following lemma shows how λ_f changes after one iteration of Algorithm 4.2.1.

Lemma 4.2.2. *Let $k \geq 0$, and let $\eta_k \geq 1$ be such that*

$$G_k \preceq \eta_k A. \tag{4.2.5}$$

Then,

$$\lambda_f(x_{k+1}) \leq (1 - \eta_k^{-1}) \lambda_f(x_k) = \frac{\eta_k - 1}{\eta_k} \lambda_f(x_k).$$

Proof. Using the fact that f is a quadratic function and substituting the

definition of x_{k+1} from Algorithm 4.2.1, we obtain

$$\nabla f(x_{k+1}) = \nabla f(x_k) + A(x_{k+1} - x_k) = A(A^{-1} - G_k^{-1})\nabla f(x_k).$$

Therefore, in view of (4.2.4),

$$\lambda_f(x_{k+1}) = \langle \nabla f(x_k), (A^{-1} - G_k^{-1})A(A^{-1} - G_k^{-1})\nabla f(x_k)\rangle^{1/2}.$$

According to (4.2.5) and (4.2.3), we have

$$\eta_k^{-1} A^{-1} \preceq G_k^{-1} \preceq A^{-1}.$$

Hence,

$$0 \preceq A^{-1} - G_k^{-1} \preceq (1 - \eta_k^{-1})A^{-1}. \tag{4.2.6}$$

Consequently,

$$(A^{-1} - G_k^{-1})A(A^{-1} - G_k^{-1}) \preceq (1 - \eta_k^{-1})^2 A^{-1},$$

and, in view of (4.2.6) and (4.2.4),

$$\lambda_f(x_{k+1}) \leq (1 - \eta_k^{-1})\langle \nabla f(x_k), A^{-1}\nabla f(x_k)\rangle^{1/2} = (1 - \eta_k^{-1})\lambda_f(x_k). \quad \square$$

Thus, to estimate how fast $\lambda_f(x_k)$ converges to zero, we need to upper bound η_k. There are two ways to proceed, depending on the choice of directions u_k in Algorithm 4.2.1.

First, consider the general situation, when we do not impose any restrictions on u_k. In this case, we can guarantee that η_k stays uniformly bounded, and $\lambda_f(x_k) \to 0$ at a *linear* rate.

Theorem 4.2.3. *For all $k \geq 0$, in Algorithm 4.2.1, we have*

$$A \preceq G_k \preceq \varkappa A, \tag{4.2.7}$$

and

$$\lambda_f(x_k) \leq (1 - \varkappa^{-1})^k \lambda_f(x_0). \tag{4.2.8}$$

Proof. Since $G_0 = LB$, in view of (4.2.2), we have

$$A \preceq G_0 \preceq \varkappa A.$$

By Lemma 4.1.2, this implies (4.2.7). Applying now Lemma 4.2.2, we obtain

$$\lambda_f(x_{k+1}) \leq (1 - \varkappa^{-1})\lambda_f(x_k)$$

for all $k \geq 0$, and (4.2.8) follows. □

Note that the right-hand side in (4.2.8) is exactly the convergence rate of the standard Gradient Method. Thus, according to Theorem 4.2.3, the convergence rate of Algorithm 4.2.1 is at least as good as that of the Gradient Method.

Now assume that the directions u_k in Algorithm 4.2.1 are chosen in accordance with the greedy strategy (4.1.10). Recall that, in this case, we can guarantee that $G_k \to A$ (Theorem 4.1.5). Therefore, we can expect faster convergence from Algorithm 4.2.1.

Theorem 4.2.4. *Suppose that, for each $k \geq 0$, we choose $u_k = \bar{u}_A(G_k)$ in Algorithm 4.2.1. Then, for all $k \geq 0$, we have*

$$A \preceq G_k \preceq \left[1 + \left(1 - (n\varkappa)^{-1}\right)^k n\varkappa\right] A, \tag{4.2.9}$$

and

$$\lambda_f(x_{k+1}) \leq \left(1 - (n\varkappa)^{-1}\right)^k n\varkappa\lambda_f(x_k). \tag{4.2.10}$$

Proof. We already know that $A \preceq G_k$. Hence, all the eigenvalues of $G_k - A$ w.r.t. A are nonnegative. Bounding the maximal one via the sum of all others (see Proposition 2.1.3(iv)), we obtain

$$G_k - A \preceq \langle A^{-1}, G_k - A\rangle A \overset{(4.1.8)}{=} \sigma_A(G_k)A,$$

or, equivalently,

$$G_k \preceq \left(1 + \sigma_A(G_k)\right)A.$$

At the same time, by Theorem 4.1.5, we have

$$\sigma_A(G_k) \leq \left(1 - (n\varkappa)^{-1}\right)^k \sigma_A(G_0).$$

Note that

$$\begin{aligned}\sigma_A(G_0) &\overset{(4.1.8)}{=} \langle A^{-1}, G_0\rangle - n \\ &\overset{(4.2.7)}{\leq} \langle A^{-1}, \varkappa A\rangle - n \overset{(2.1.25)}{=} n(\varkappa - 1) \leq n\varkappa.\end{aligned}$$

This proves (4.2.9). Applying now Lemma 4.2.2 and using the fact that $\frac{\eta-1}{\eta} \leq \eta - 1$ for any $\eta \geq 1$, we obtain (4.2.10). □

Theorem 4.2.4 shows that the convergence rate of $\lambda_f(x_k)$ is *superlinear*. Let us now combine this result with Theorem 4.2.3 and write down the final efficiency estimate. Denote by $k_0 \geq 0$ the number of the first iteration for which

$$\left(1 - (n\varkappa)^{-1}\right)^{k_0} n\varkappa \leq \frac{1}{2}. \tag{4.2.11}$$

Clearly,

$$k_0 \leq \lceil n\varkappa \ln(2n\varkappa) \rceil.$$

According to Theorem 4.1.5, during the first k_0 iterations,

$$\lambda_f(x_k) \leq (1 - \varkappa^{-1})^k \lambda_f(x_0). \tag{4.2.12}$$

After that, by Theorem 4.2.4, for all $k \geq 0$, we have

$$\begin{aligned} \lambda_f(x_{k_0+k+1}) &\overset{(4.2.10)}{\leq} \left(1 - (n\varkappa)^{-1}\right)^{k_0+k} n\varkappa \lambda_f(x_{k_0+k}) \\ &\overset{(4.2.11)}{\leq} \left(1 - (n\varkappa)^{-1}\right)^{k} \frac{1}{2} \lambda_f(x_{k_0+k}). \end{aligned}$$

Thus, for all $k \geq 0$,

$$\begin{aligned} \lambda_f(x_{k_0+k}) &\leq \lambda_f(x_{k_0}) \prod_{i=0}^{k-1} \left[\left(1 - (n\varkappa)^{-1}\right)^i \frac{1}{2}\right] \\ &= \left(1 - (n\varkappa)^{-1}\right)^{\sum_{i=0}^{k-1} i} \left(\frac{1}{2}\right)^k \lambda_f(x_{k_0}) \\ &= \left(1 - (n\varkappa)^{-1}\right)^{k(k-1)/2} \left(\frac{1}{2}\right)^k \lambda_f(x_{k_0}) \\ &\overset{(4.2.12)}{\leq} \left(1 - (n\varkappa)^{-1}\right)^{k(k-1)/2} \left(\frac{1}{2}\right)^k (1 - \varkappa^{-1})^{k_0} \lambda_f(x_0). \end{aligned}$$

Note that the first factor in this estimate depends on the *square* of the iteration counter.

To conclude, let us mention one important property of Algorithm 4.2.1 with greedily selected u_k. It turns out that, in the particular case $\chi_k \equiv 0$, i.e., when Algorithm 4.2.1 corresponds to the Greedy *SR1* Method, it will identify the operator A, and consequently, the minimizer x^* of the quadratic function (4.2.1), in a *finite* number of steps.

Theorem 4.2.5. *Suppose that, in Algorithm 4.2.1, for each $k \geq 0$, we choose $u_k = \bar{u}_A(G_k)$ and $\chi_k = 0$. Then $G_k = A$ for some $0 \leq k \leq n$.*

Proof. Suppose that $R_k \coloneqq G_k - A \neq 0$ for all $0 \leq k \leq n$. Since $R_k \succeq 0$ (see (4.2.3)), in view of (4.1.10), we must have $u_k \notin \ker R_k$, and, according to (4.1.3),

$$R_{k+1} = R_k - \frac{R_k u_k u_k^* R_k}{\langle R_k u_k, u_k \rangle}$$

for all $0 \leq k \leq n$. From this formula, it is easily seen that

(1) $\ker R_k \subseteq \ker R_{k+1}$,

(2) $u_k \in \ker R_{k+1}$.

Thus, the dimension of $\ker R_k$ grows at least by 1 at every iteration. In particular, the dimension of $\ker R_{n+1}$ must be at least $n+1$, which is impossible, since the operator R_{n+1} acts in an n-dimensional vector space. □

It is worth noting that for other updates (e.g., DFP or BFGS), the inclusion $\ker R_k \subseteq \ker R_{k+1}$ is, in general, no longer valid.

4.3 Minimization of General Functions

Now consider a general problem of unconstrained minimization:

$$\min_{x \in \mathbb{E}} f(x), \tag{4.3.1}$$

where $f \colon \mathbb{E} \to \mathbb{R}$ is a twice differentiable function with positive definite Hessian. Our goal is to extend the results obtained in the previous section to the problem (4.3.1), assuming that the methods can start from a sufficiently good initial point x_0.

We make the same assumptions about the objective function f as in Chapter 3. Namely, we assume that f is strongly convex, strongly self-concordant and its gradient is Lipschitz continuous, i.e., there exist $\mu, L > 0$ and $M \geq 0$ such that, for all $x, y, z, w \in \mathbb{E}$, we have

$$\mu B \preceq \nabla^2 f(x) \preceq LB, \tag{4.3.2}$$

$$\nabla^2 f(x) - \nabla^2 f(y) \preceq M \|x - y\|_z \, \nabla^2 f(w). \tag{4.3.3}$$

The only difference compared to Chapter 3 is that the operator B in (4.3.2) cannot be arbitrary and must now coincide with the one from (4.1.11).

Recall that the ratio of the constants L and μ is called the *condition number* of problem (4.3.1):

$$\varkappa := \frac{L}{\mu} \ (\geq 1).$$

Remark 4.3.1. In fact, for our purposes, it is enough to require that (4.3.2) and (4.3.3) hold only in a neighborhood of a solution, but, for the sake of simplicity, we do not do this.

Let us now estimate the progress of a general quasi-Newton step. As before, for measuring the progress, we use the *local norm of the gradient*:

$$\lambda_f(x) := \|\nabla f(x)\|_x^* = \langle \nabla f(x), [\nabla^2 f(x)]^{-1} \nabla f(x) \rangle^{1/2}, \qquad x \in \mathbb{E}. \quad (4.3.4)$$

Lemma 4.3.2. *Let $x \in \mathbb{E}$, $G \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ and $\eta \geq 1$ be such that*

$$\nabla^2 f(x) \preceq G \preceq \eta \nabla^2 f(x). \quad (4.3.5)$$

Let

$$x_+ := x - G^{-1} \nabla f(x), \quad (4.3.6)$$

and let $\lambda := \lambda_f(x)$ be such that $M\lambda \leq 2$. Then, $r := \|x_+ - x\|_x \leq \lambda$, and

$$\lambda_f(x_+) \leq (1 + \tfrac{1}{2} M\lambda)(\eta - 1 + \tfrac{1}{2} M\lambda)\eta^{-1}\lambda. \quad (4.3.7)$$

Proof. Denote $J := \int_0^1 \nabla^2 f\big(x + t(x_+ - x)\big) dt$. Applying Taylor's formula and using (4.3.6), we obtain

$$\nabla f(x_+) = \nabla f(x) + J(x_+ - x) = J(J^{-1} - G^{-1}) \nabla f(x). \quad (4.3.8)$$

Note that

$$\begin{aligned} r &= \|x_+ - x\|_x \overset{(4.3.6)}{=} \|G^{-1} \nabla f(x)\|_x \\ &= \langle \nabla f(x), G^{-1} \nabla^2 f(x) G^{-1} \nabla f(x) \rangle^{1/2} \overset{(4.3.5)}{\leq} \langle \nabla f(x), G^{-1} \nabla f(x) \rangle^{1/2} \\ &\overset{(4.3.5)}{\leq} \langle \nabla f(x), \nabla^2 f(x)^{-1} \nabla f(x) \rangle^{1/2} \overset{(4.3.4)}{=} \lambda. \end{aligned}$$

Hence, in view of Lemma 3.3.5, we have

$$(1 + \tfrac{1}{2} M\lambda)^{-1} \nabla^2 f(x) \preceq J \preceq (1 + \tfrac{1}{2} M\lambda) \nabla^2 f(x), \quad (4.3.9)$$

$$J \preceq (1 + \tfrac{1}{2} M\lambda) \nabla^2 f(x_+). \quad (4.3.10)$$

Therefore, according to (4.3.4) and (4.3.8),

$$\begin{aligned}\lambda_f^2(x_+) &= \langle \nabla f(x_+), \nabla^2 f(x_+)^{-1} \nabla f(x_+) \rangle \\ &\le (1 + \tfrac{1}{2} M\lambda) \langle \nabla f(x_+), J^{-1} \nabla f(x_+) \rangle \\ &= (1 + \tfrac{1}{2} M\lambda) \langle \nabla f(x), (J^{-1} - G^{-1}) J (J^{-1} - G^{-1}) \nabla f(x) \rangle.\end{aligned} \tag{4.3.11}$$

Further, by (4.3.9) and (4.3.5), we have

$$(1 + \tfrac{1}{2} M\lambda)^{-1} J \preceq \nabla^2 f(x) \preceq G \preceq \eta \nabla^2 f(x) \preceq \eta (1 + \tfrac{1}{2} M\lambda) J.$$

Hence,

$$[(1 + \tfrac{1}{2} M\lambda)\eta]^{-1} J^{-1} \preceq G^{-1} \preceq (1 + \tfrac{1}{2} M\lambda) J^{-1},$$

and

$$-\bigl(1 - [(1 + \tfrac{1}{2} M\lambda)\eta]^{-1}\bigr) J^{-1} \preceq G^{-1} - J^{-1} \preceq \tfrac{1}{2} M\lambda J^{-1}.$$

Note that

$$1 - [(1 + \tfrac{1}{2} M\lambda)\eta]^{-1} \le 1 - (1 - \tfrac{1}{2} M\lambda)\eta^{-1} = (\eta - 1 + \tfrac{1}{2} M\lambda)\eta^{-1},$$

and, since $M\lambda \le 2$ and $\eta \ge 1$,

$$\tfrac{1}{2} M\lambda = 1 - (1 - \tfrac{1}{2} M\lambda) \le 1 - (1 - \tfrac{1}{2} M\lambda)\eta^{-1} = (\eta - 1 + \tfrac{1}{2} M\lambda)\eta^{-1}.$$

Therefore,

$$-(\eta - 1 + \tfrac{1}{2} M\lambda)\eta^{-1} J^{-1} \preceq G^{-1} - J^{-1} \preceq (\eta - 1 + \tfrac{1}{2} M\lambda)\eta^{-1} J^{-1}.$$

Consequently,

$$(G^{-1} - J^{-1}) J (G^{-1} - J^{-1}) \preceq \bigl((\eta - 1 + \tfrac{1}{2} M\lambda)\eta^{-1}\bigr)^2 J^{-1}.$$

Combining this with (4.3.11) and (4.3.9), we obtain

$$\begin{aligned}\lambda_f(x_+) &\le \sqrt{1 + \tfrac{1}{2} M\lambda}\, (\eta - 1 + \tfrac{1}{2} M\lambda)\eta^{-1} \langle \nabla f(x), J^{-1} \nabla f(x) \rangle^{1/2} \\ &\le (1 + \tfrac{1}{2} M\lambda)(\eta - 1 + \tfrac{1}{2} M\lambda)\eta^{-1} \langle \nabla f(x), \nabla^2 f(x)^{-1} \nabla f(x) \rangle^{1/2} \\ &= (1 + \tfrac{1}{2} M\lambda)(\eta - 1 + \tfrac{1}{2} M\lambda)\eta^{-1} \lambda.\end{aligned}$$

This is exactly (4.3.7). □

Now we need to analyze what happens with the Hessian approxima-

tion after a quasi-Newton update. Let G be the current approximation of $\nabla^2 f(x)$, satisfying, as usual, the condition

$$\nabla^2 f(x) \preceq G. \tag{4.3.12}$$

Using this approximation, we can compute the new test point

$$x_+ = x - G^{-1}\nabla f(x).$$

After that, we would like to update G into a new operator G_+, approximating the Hessian $\nabla^2 f(x_+)$ at the new point and satisfying the condition

$$\nabla^2 f(x_+) \preceq G_+.$$

A natural idea is, of course, to set

$$G_+ = \mathrm{EBroyd}_\chi\bigl(\nabla^2 f(x_+), G, u\bigr) \tag{4.3.13}$$

for some $u \in \mathbb{E}$ and $\chi \in [0,1]$. However, we cannot do this, since the update (4.3.13) is well-defined only when

$$\nabla^2 f(x_+) \preceq G$$

(see Section 4.1), which may not be true, even though (4.3.12) holds. To avoid this problem, let us apply the following *correction strategy*:

1. Choose some $\delta \geq 0$, and set $\tilde{G} = (1+\delta)G$.
2. Compute G_+, using (4.3.13) with G replaced by $\tilde{G}$.

Clearly, for a sufficiently large value of δ, the condition $\nabla^2 f(x_+) \preceq \tilde{G}$ will be valid. If, at the same time, this δ is sufficiently small, then the above correction strategy should not introduce too big an error.

Lemma 4.3.3. *Let $x \in \mathbb{E}$, $G \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ and $\eta \geq 1$ be such that*

$$\nabla^2 f(x) \preceq G \preceq \eta \nabla^2 f(x). \tag{4.3.14}$$

Further, let $x_+ \in \mathbb{E}$ and $r \coloneqq \|x_+ - x\|_x$. Then

$$\tilde{G} \coloneqq (1+Mr)G \succeq \nabla^2 f(x_+), \tag{4.3.15}$$

and, for all $u \in \mathbb{E}$ *and* $\chi \in [0,1]$*, we have*

$$\nabla^2 f(x_+) \preceq \mathrm{EBroyd}_\chi\big(\nabla^2 f(x_+), \tilde{G}, u\big) \preceq [(1+Mr)^2\eta]\nabla^2 f(x_+),$$

Proof. According to Lemma 3.3.5 and (4.3.14), we have

$$\begin{aligned}\nabla^2 f(x_+) &\preceq (1+Mr)\nabla^2 f(x) \preceq (1+Mr)G = \tilde{G},\\ \tilde{G} = (1+Mr)G &\preceq (1+Mr)\eta\nabla^2 f(x) \preceq (1+Mr)^2\eta\nabla^2 f(x_+).\end{aligned}$$

Thus,

$$\nabla^2 f(x_+) \preceq \tilde{G} \preceq (1+Mr)^2\eta\nabla^2 f(x_+),$$

and the claim now follows from Lemma 4.1.2. □

We are ready to write down the scheme of our quasi-Newton methods. For simplicity, we assume that the constants L and M from (4.3.2) and (4.3.3) are known.

Algorithm 4.3.1: Extended Convex Broyden Method
Initialization: Choose $x_0 \in \mathbb{E}$. Set $G_0 = LB$.
For $k \geq 0$ **iterate:** 1. Update $x_{k+1} = x_k - G_k^{-1}\nabla f(x_k)$. 2. Compute $r_k = \|x_{k+1} - x_k\|_{x_k}$ and set $\tilde{G}_k = (1+Mr_k)G_k$. 3. Choose $u_k \in \mathbb{E}$ and $\chi_k \in [0,1]$. 4. Compute $G_{k+1} = \mathrm{EBroyd}_{\chi_k}\big(\nabla^2 f(x_{k+1}), \tilde{G}_k, u_k\big)$.

Remark 4.3.4. For the moment, we do not impose any restrictions on the choice of updating directions u_k in Algorithm 4.3.1. However, eventually, we will assume that u_k are chosen in accordance with the greedy strategy.

Remark 4.3.5. As in Remark 4.2.1, in an actual implementation of Algorithm 4.3.1, one should work directly with $H_k := G_k^{-1}$ in order to keep the iteration cost low. Note also that, for implementing the corresponding inverse Hessian approximation update at Step 4, one needs to compute the Hessian-vector product $\nabla^2 f(x_{k+1})u_k$. This is in contrast to classical quasi-Newton methods (see Algorithm 3.4.1), for which we only need the gradients. However, this is not a big issue since, for the majority of functions, arising in real-life applications, the Hessian-vector product can be

efficiently computed at basically the same cost as the gradient (e.g., by automatic differentiation or finite differences).

As before, we present two convergence results for Algorithm 4.3.1. The first one establishes linear convergence and can be seen as a generalization of Theorem 4.2.3. Note that for this result the directions u_k in Algorithm 4.3.1 can be chosen arbitrarily.

Theorem 4.3.6. *Suppose the initial point x_0 is sufficiently close to the solution:*

$$M\lambda_f(x_0) \leq \frac{\ln(3/2)}{4}\varkappa^{-1}. \tag{4.3.16}$$

Then, for all $k \geq 0$, we have

$$\nabla^2 f(x_k) \preceq G_k \preceq \exp\Bigl(2M\sum_{i=0}^{k-1}\lambda_f(x_i)\Bigr)\varkappa\nabla^2 f(x_k) \preceq \frac{3}{2}\varkappa\nabla^2 f(x_k), \tag{4.3.17}$$

and

$$\lambda_f(x_k) \leq \bigl(1-(2\varkappa)^{-1}\bigr)^k \lambda_f(x_0). \tag{4.3.18}$$

Proof. In view of (4.3.2), we have

$$\nabla^2 f(x_0) \preceq G_0 \preceq \varkappa\nabla^2 f(x_0).$$

Therefore, for $k = 0$, both (4.3.17) and (4.3.18) are satisfied.

Now let $k \geq 0$, and suppose (4.3.17) and (4.3.18) have already been proved for all $0 \leq k' \leq k$. Denote $\lambda_k := \lambda_f(x_k)$, $r_k := \|x_{k+1} - x_k\|_{x_k}$, and

$$\eta_k := \exp\Bigl(2M\sum_{i=0}^{k-1}\lambda_i\Bigr)\varkappa. \tag{4.3.19}$$

Note that, according to (4.3.18) and (4.3.16),

$$M\sum_{i=0}^{k}\lambda_i \leq M\lambda_0\sum_{i=0}^{k}\bigl(1-(2\varkappa)^{-1}\bigr)^i \leq 2\varkappa M\lambda_0 \leq \frac{\ln(3/2)}{2}. \tag{4.3.20}$$

Applying Lemma 4.3.2, we obtain that

$$r_k \leq \lambda_k \tag{4.3.21}$$

and

$$\begin{aligned}\lambda_{k+1} &\le (1+\tfrac{1}{2}M\lambda_k)(\eta_k - 1 + \tfrac{1}{2}M\lambda_k)\eta_k^{-1}\lambda_k \\ &= (1+\tfrac{1}{2}M\lambda_k)\big(1-(1-\tfrac{1}{2}M\lambda_k)\eta_k^{-1}\big)\lambda_k.\end{aligned} \tag{4.3.22}$$

Using the fact that $1 - t \ge \exp(-2t)$ for any $0 \le t \le \frac{1}{2}$, we obtain

$$\begin{aligned}(1-\tfrac{1}{2}M\lambda_k)\eta_k^{-1} &\ge \exp(-M\lambda_k)\eta_k^{-1} \\ &\overset{(4.3.19)}{=} \exp\Big(-M\lambda_k - 2M\sum_{i=0}^{k-1}\lambda_i\Big)\varkappa^{-1} \\ &\ge \exp\Big(-2M\sum_{i=0}^{k}\lambda_i\Big)\varkappa^{-1} \overset{(4.3.20)}{\ge} \frac{2}{3}\varkappa^{-1}.\end{aligned}$$

Also, since $\ln(1+t) \le t$ for any $t \ge 0$, we obtain from (4.3.16) that

$$\frac{1}{2}M\lambda_k \le \frac{\ln(3/2)}{8}\varkappa^{-1} \le (16\varkappa)^{-1}.$$

Hence,

$$\begin{aligned}&(1+\tfrac{1}{2}M\lambda_k)\big(1-(1-\tfrac{1}{2}M\lambda_k)\eta_k^{-1}\big) \\ &\quad\le \big(1+(16\varkappa)^{-1}\big)\Big(1-\frac{2}{3}\varkappa^{-1}\Big) \\ &\quad\le 1-\Big(\frac{2}{3}-\frac{1}{16}\Big)\varkappa^{-1} \le 1-(2\varkappa)^{-1}.\end{aligned}$$

Consequently, according to (4.3.22) and (4.3.18),

$$\lambda_{k+1} \le \big(1-(2\varkappa)^{-1}\big)\lambda_k \le \big(1-(2\varkappa)^{-1}\big)^{k+1}\lambda_0.$$

Finally, from Lemma 4.3.3, it follows that

$$\begin{aligned}\nabla^2 f(x_{k+1}) &\preceq G_{k+1} \preceq (1+Mr_k)^2\eta_k\nabla^2 f(x_{k+1}) \\ &\overset{(4.3.21)}{\preceq} (1+M\lambda_k)^2\eta_k\nabla^2 f(x_{k+1}) \preceq \exp(2M\lambda_k)\eta_k\nabla^2 f(x_{k+1}) \\ &\overset{(4.3.19)}{=} \exp\Big(2M\sum_{i=0}^{k}\lambda_i\Big)\varkappa\nabla^2 f(x_{k+1}) \overset{(4.3.20)}{\preceq} \frac{3}{2}\varkappa\nabla^2 f(x_{k+1}).\end{aligned}$$

Thus, (4.3.17) and (4.3.18) are valid for $k' = k+1$, and we can continue by induction. □

Now let us analyze the greedy strategy. First, we analyze how the Hes-

sian approximation measure (4.1.8) changes after one iteration. In what follows, for the sake of convenience, for any $x \in \mathbb{E}$ and any $G \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$, we use the following shortcut:

$$\sigma_x(G) \coloneqq \sigma_{\nabla^2 f(x)}(G).$$

Lemma 4.3.7. *Let $x \in \mathbb{E}$ and $G \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$ be such that $\nabla^2 f(x) \preceq G$. Further, let $x_+ \in \mathbb{E}$, $r \coloneqq \|x_+ - x\|_x$ and*

$$\tilde{G} \coloneqq (1 + Mr)G. \tag{4.3.23}$$

Then, for any $\chi \in [0, 1]$ and $G_+ \coloneqq \mathrm{EBroyd}_\chi\big(\nabla^2 f(x_+), \tilde{G}, \bar{u}_{x_+}(G)\big)$, we have

$$\sigma_{x_+}(G_+) \leq \big(1 - (n\varkappa)^{-1}\big)(1 + Mr)^2\Big(\sigma_x(G) + \frac{2nMr}{1 + Mr}\Big).$$

Proof. We already know from Lemma 4.3.3 that $\nabla^2 f(x_+) \preceq \tilde{G}$. Also note that $\bar{u}_{x_+}(\tilde{G}) = \bar{u}_{x_+}(G)$ (see (4.1.10)). Hence, by Theorem 4.1.5, we have

$$\sigma_{x_+}(G_+) \leq \big(1 - (n\varkappa)^{-1}\big)\sigma_{x_+}(\tilde{G}).$$

Using (4.1.8) and (4.3.23) and Lemma 3.3.5, we further get

$$\begin{aligned}
\sigma_{x_+}(\tilde{G}) &= \langle \nabla^2 f(x_+)^{-1}, \tilde{G}\rangle - n \\
&= (1 + Mr)\langle \nabla^2 f(x_+)^{-1}, G\rangle - n \\
&\leq (1 + Mr)^2\langle \nabla^2 f(x)^{-1}, G\rangle - n \\
&= (1 + Mr)^2\big(\sigma_x(G) + n\big) - n \\
&= (1 + Mr)^2\sigma_x(G) + n\big((1 + Mr)^2 - 1\big) \\
&= (1 + Mr)^2\sigma_x(G) + 2nMr(1 + \tfrac{1}{2}Mr) \\
&\leq (1 + Mr)^2\Big(\sigma_x(G) + \frac{2nMr}{1 + Mr}\Big).
\end{aligned}$$

Putting everything together, we obtain the claim. □

Now we can prove superlinear convergence. In what follows, we assume that $n \geq 2$.

Theorem 4.3.8. *Suppose that, in Algorithm 4.3.1, for each $k \geq 0$, we take $u_k = \bar{u}_{x_{k+1}}(G_k)$. Also, suppose that the initial point x_0 is sufficiently close*

to the solution:

$$M\lambda_f(x_0) \leq \frac{\ln 2}{4(2n+1)}\varkappa^{-1} \left(\leq \frac{\ln(3/2)}{4}\varkappa^{-1}\right). \tag{4.3.24}$$

Then, for all $k \geq 0$, we have

$$\nabla^2 f(x_k) \preceq G_k \preceq \left[1 + \left(1 - (n\varkappa)^{-1}\right)^k 2n\varkappa\right]\nabla^2 f(x_k), \tag{4.3.25}$$

and

$$\lambda_f(x_{k+1}) \leq \left(1 - (n\varkappa)^{-1}\right)^k 2n\varkappa\lambda_f(x_k). \tag{4.3.26}$$

Proof. Denote $\lambda_k \coloneqq \lambda_f(x_k)$ and $\sigma_k \coloneqq \sigma_{x_k}(G_k)$ for $k \geq 0$. In view of Theorem 4.3.6, the first relation in (4.3.25) is indeed true, and also

$$M\sum_{i=0}^{k}\lambda_i \leq M\lambda_0\sum_{i=0}^{k}\left(1-(2\varkappa)^{-1}\right)^i \leq 2\varkappa\lambda_0 \overset{(4.3.24)}{\leq} \frac{\ln 2}{2(2n+1)} \tag{4.3.27}$$

for all $k \geq 0$.

Let us show by induction that, for all $k \geq 0$, we have

$$\sigma_k + 2nM\lambda_k \leq \theta_k, \tag{4.3.28}$$

where

$$\begin{aligned}\theta_k &\coloneqq \left(1-(n\varkappa)^{-1}\right)^k \exp\Big(2(2n+1)M\sum_{i=0}^{k-1}\lambda_i\Big)n\varkappa \\ &\leq \left(1-(n\varkappa)^{-1}\right)^k 2n\varkappa.\end{aligned} \tag{4.3.29}$$

(The inequality follows from (4.3.27)). Indeed, according to (4.3.2), we have $\nabla^2 f(x_0) \preceq G_0 \preceq \varkappa\nabla^2 f(x_0)$. Hence,

$$\begin{aligned}\sigma_0 + 2nM\lambda_0 &\overset{(4.1.8)}{=} \langle \nabla^2 f(x_0)^{-1}, G_0\rangle - n + 2nM\lambda_0 \\ &\leq \langle \nabla^2 f(x_0)^{-1}, \varkappa\nabla^2 f(x_0)\rangle - n + 2nM\lambda_0 \\ &\overset{(2.1.25)}{=} n(\varkappa - 1) + 2nM\lambda_0 \\ &\overset{(4.3.24)}{\leq} n(\varkappa-1) + \frac{n\ln 2}{2(2n+1)} \leq n\varkappa.\end{aligned}$$

Therefore, for $k = 0$, (4.3.28) is satisfied. Now suppose that it is also

satisfied for some $k \geq 0$. Since $\nabla^2 f(x_k) \preceq G_k$, all the eigenvalues of $G_k - \nabla^2 f(x_k)$ w.r.t. $\nabla^2 f(x_k)$ are nonnegative. Bounding the maximal one via the sum of the others, we obtain

$$G_k - \nabla^2 f(x_k) \preceq \sigma_k \nabla^2 f(x_k),$$

or, equivalently,

$$G_k \preceq (1 + \sigma_k) \nabla^2 f(x_k). \tag{4.3.30}$$

Therefore, applying Lemma 4.3.2, we obtain

$$r_k := \|x_{k+1} - x_k\|_{x_k} \leq \lambda_k, \tag{4.3.31}$$

and

$$\begin{aligned} \lambda_{k+1} &\leq (1 + \tfrac{1}{2} M\lambda_k)(\sigma_k + \tfrac{1}{2} M\lambda_k)(1 + \sigma_k)^{-1} \lambda_k \\ &\leq (1 + \tfrac{1}{2} M\lambda_k)(\sigma_k + 2nM\lambda_k)\lambda_k \leq (1 + \tfrac{1}{2} M\lambda_k)\theta_k \lambda_k \\ &\leq \exp(\tfrac{1}{2} M\lambda_k)\theta_k \lambda_k \leq \exp(2M\lambda_k)\theta_k \lambda_k, \end{aligned} \tag{4.3.32}$$

where the third inequality follows from (4.3.28). Further, by Lemma 4.3.7,

$$\begin{aligned} \sigma_{k+1} &\leq \big(1 - (n\varkappa)^{-1}\big)(1 + Mr_k)^2 \Big(\sigma_k + \frac{2nMr_k}{1 + Mr_k}\Big) \\ &\overset{(4.3.31)}{\leq} \big(1 - (n\varkappa)^{-1}\big)(1 + M\lambda_k)^2 \Big(\sigma_k + \frac{2nM\lambda_k}{1 + M\lambda_k}\Big) \\ &\leq \big(1 - (n\varkappa)^{-1}\big)(1 + M\lambda_k)^2 (\sigma_k + 2nM\lambda_k) \\ &\overset{(4.3.28)}{\leq} \big(1 - (n\varkappa)^{-1}\big)(1 + M\lambda_k)^2 \theta_k \\ &\leq \big(1 - (n\varkappa)^{-1}\big) \exp(2M\lambda_k)\theta_k. \end{aligned}$$

Note that $\frac{1}{2} \leq 1 - (n\varkappa)^{-1}$ since $n \geq 2$. Therefore,

$$\begin{aligned} &\sigma_{k+1} + 2nM\lambda_{k+1} \\ &\quad \leq \big(1 - (n\varkappa)^{-1}\big) \exp(2M\lambda_k)\theta_k + \exp(2M\lambda_k)\theta_k \cdot 2nM\lambda_k \\ &\quad \leq \big(1 - (n\varkappa)^{-1}\big) \exp(2M\lambda_k)\theta_k + \big(1 - (n\varkappa)^{-1}\big) \exp(2M\lambda_k)\theta_k \cdot 4nM\lambda_k \\ &\quad = \big(1 - (n\varkappa)^{-1}\big) \exp(2M\lambda_k)(1 + 4nM\lambda_k)\theta_k \\ &\quad \leq \big(1 - (n\varkappa)^{-1}\big) \exp\big(2(2n+1)M\lambda_k\big)\theta_k = \theta_{k+1}, \end{aligned}$$

where the last identity is due to (4.3.29). Thus, (4.3.28) is proved.

Let us fix now some $k \geq 0$. Since $\lambda_k \geq 0$, we have, according to (4.3.28) and (4.3.29),

$$\sigma_k \leq \sigma_k + 2M\lambda_k \leq \theta_k \leq \left(1 - (n\varkappa)^{-1}\right)^k 2n\varkappa.$$

This proves the second relation in (4.3.25) in view of (4.3.30). Finally, combining (4.3.32) and (4.3.29), we obtain

$$\begin{aligned}\lambda_{k+1} &\leq \exp(2M\lambda_k)\theta_k\lambda_k \leq \exp\bigl(2(2n+1)M\lambda_k\bigr)\theta_k\lambda_k \\ &= \left(1 - (n\varkappa)^{-1}\right)^{-1}\theta_{k+1}\lambda_k \leq \left(1 - (n\varkappa)^{-1}\right)^k 2n\varkappa\lambda_k,\end{aligned}$$

which proves (4.3.26). □

As in the quadratic case, combining Theorems 4.3.6 and 4.3.8, we obtain the following efficiency estimate, for all $k \geq 0$:

$$\lambda_f(x_{k_0+k}) \leq \left(1 - (n\varkappa)^{-1}\right)^{k(k-1)/2} 2^{-k} \left(1 - (2\varkappa)^{-1}\right)^{k_0} \lambda_f(x_0),$$

where

$$k_0 := \lceil n\varkappa \ln(2n\varkappa) \rceil.$$

4.4 Comparison with Classical Methods

Let us compare the rates of superlinear convergence we have obtained for the greedy quasi-Newton methods with those of the classical ones from Chapter 3. For brevity, we discuss only the DFP and BFGS methods. Furthermore, since the estimates for the general nonlinear case differ from those for the quadratic one only in absolute constants (both for the greedy and classical methods), we only consider the case when the objective function is quadratic.

We use our standard notation: n is the dimension of the space, μ is the strong convexity parameter, L is the Lipschitz constant of the gradient, and λ_k is the local norm of the gradient at the kth iteration. Further, to avoid some technicalities and keep the presentation simple, we assume that the *condition number* of the problem is not especially good, namely,

$$\varkappa := \frac{L}{\mu} \geq 3. \tag{4.4.1}$$

For the greedy quasi-Newton methods (both DFP and BFGS), we have

the following recurrence, for all $k \geq 0$ (see Theorem 4.2.4):

$$\lambda_{k+1} \leq \bigl(1 - (n\varkappa)^{-1}\bigr)^k n\varkappa\lambda_k \leq \exp\bigl(-k/(n\varkappa)\bigr)n\varkappa\lambda_k. \tag{4.4.2}$$

Thus, their rate of superlinear convergence is described by the inequality

$$\begin{aligned}\lambda_k &\leq \lambda_0 \prod_{i=0}^{k-1} \bigl[\exp\bigl(-k/(n\varkappa)\bigr)n\varkappa\bigr] \\ &= \exp\bigl(-\tfrac{1}{2}k(k-1)/(n\varkappa)\bigr)(n\varkappa)^k\lambda_0.\end{aligned} \tag{4.4.3}$$

This inequality is formally valid for all $k \geq 1$. However, it is useful only when the factor in front of λ_0 is smaller than or equal to 1, i.e., when $k \geq K_0^{\mathrm{Gr}}$, where

$$K_0^{\mathrm{Gr}} := 1 + \lceil 2n\varkappa \ln(n\varkappa)\rceil. \tag{4.4.4}$$

The number K_0^{Gr} is the *starting moment* of superlinear convergence of the greedy DFP and BFGS methods, according to the estimate (4.4.3).

For the classical DFP Method, we have the following bound, for all $k \geq 1$ (Theorem 3.2.3 with $\chi_i \equiv 1$):

$$\lambda_k \leq \bigl[2\varkappa(\varkappa^{n/k} - 1)\bigr]^{k/2}\sqrt{\varkappa}\,\lambda_0, \tag{4.4.5}$$

and the starting moment of superlinear convergence is of the order

$$K_0^{\mathrm{DFP}} \sim n\varkappa \ln \varkappa \tag{4.4.6}$$

(see the corresponding discussion after Theorem 3.2.3).

Comparing the starting moments of superlinear convergence, given by (4.4.4) and (4.4.6), we see that, for the classical DFP Method, the superlinear convergence starts slightly earlier than for the greedy one. However, the difference is only in the logarithmic factor.

Nevertheless, let us show that, soon after the superlinear convergence of the Greedy DFP Method begins, namely, after

$$\tilde{K}_0^{\mathrm{Gr}} := 1 + \lceil 6n\varkappa \ln(4n\varkappa)\rceil \ (\geq 2) \tag{4.4.7}$$

iterations, it will be significantly faster than that of the classical method, according to our estimates. Indeed, denote the factors in front of λ_0 in the right-hand sides of (4.4.2) and (4.4.5) by A_k and B_k, respectively. Using

the inequality $\exp(t) \geq 1 + t$, $t \in \mathbb{R}$, and (4.4.1), we obtain, for all $k \geq 1$,

$$\begin{aligned} B_k &= \left[2\varkappa\left(\exp([n \ln \varkappa]/k) - 1\right)\right]^{k/2}\sqrt{\varkappa} \\ &\geq \left(2\frac{n\varkappa \ln \varkappa}{k}\right)^{k/2}\sqrt{\varkappa} \geq \left(\frac{n\varkappa}{k}\right)^{k/2}. \end{aligned}$$

Hence, for all $k \geq 1$,

$$\begin{aligned} \frac{A_k}{B_k} &\leq \exp\left(-\tfrac{1}{2}k(k-1)/(n\varkappa)\right)(n\varkappa)^k\left(k/(n\varkappa)\right)^{k/2} \\ &= \exp\left(-\tfrac{1}{2}k(k-1)/(n\varkappa)\right)(n\varkappa k)^{k/2}. \end{aligned} \tag{4.4.8}$$

Note that $t \mapsto \ln t/(t-1)$ is a decreasing function on $(1, +\infty)$ (since the logarithm is concave). Therefore, according to (4.4.7) and (4.4.1), for all $k \geq \tilde{K}_0^{\mathrm{Gr}}$, we have

$$\begin{aligned} \frac{n\varkappa \ln(n\varkappa k)}{k-1} &\leq \frac{n\varkappa \ln\left(n\varkappa[1 + 6n\varkappa \ln(4n\varkappa)]\right)}{6n\varkappa \ln(4n\varkappa)} \leq \frac{\ln\left(n\varkappa[1 + 24(n\varkappa)^2]\right)}{6\ln(4n\varkappa)} \\ &\leq \frac{\ln\left(48(n\varkappa)^3\right)}{6\ln(4n\varkappa)} \leq \frac{\ln\left((4n\varkappa)^3\right)}{6\ln(4n\varkappa)} = \frac{3\ln(4n\varkappa)}{6\ln(4n\varkappa)} = \frac{1}{2}. \end{aligned}$$

Consequently, for all $k \geq \tilde{K}_0^{\mathrm{Gr}}$,

$$(n\varkappa k)^{k/2} = \exp\left(\tfrac{1}{2}k\ln(n\varkappa k)\right) \leq \exp\left(\tfrac{1}{4}k(k-1)/(n\varkappa)\right).$$

Substituting this estimate into (4.4.8), we obtain, for all $k \geq \tilde{K}_0^{\mathrm{Gr}}$,

$$\frac{A_k}{B_k} \leq \exp\left(-\tfrac{1}{4}k(k-1)/(n\varkappa)\right) \leq 1.$$

Thus, after $\tilde{K}_0^{\mathrm{Gr}}$ iterations, the rate of superlinear convergence of the Greedy DFP Method is always better than that of the classical one. Moreover, as $k \to +\infty$, the gap between these rates grows as $\exp\left(O(1)k^2(n\varkappa)^{-1}\right)$.

Now let us discuss the BFGS Method. For the classical version, we have the following estimate, for all $k \geq 1$ (Theorem 3.2.3 with $\chi_i \equiv 0$):

$$\lambda_k \leq \left[2(\varkappa^{n/k} - 1)\right]^{k/2}\sqrt{\varkappa}\,\lambda_0. \tag{4.4.9}$$

The starting moment of superlinear convergence is of the order

$$K_0^{\mathrm{BFGS}} \sim n \ln \varkappa \tag{4.4.10}$$

(see the corresponding discussion after Theorem 3.2.3).

Comparing (4.4.10) with (4.4.4), we see that, for the classical BFGS Method, the starting moment of superlinear convergence is much better. It has a very weak (logarithmic) dependence on the condition number.

Nevertheless, asymptotically, the rate (4.4.9) of the classical BFGS is slower than that of the greedy one. Specifically, one can show (similarly to how this was done for DFP) that, soon after the superlinear convergence of the greedy BFGS Method begins, the corresponding rate (4.4.2) of superlinear convergence will be better than that of the classical method by a factor of $\exp\big(O(1)k^2/(n\varkappa)\big)$.

4.5 Numerical Experiments

In this section, we present preliminary computational results for greedy quasi-Newton methods and compare them with classical quasi-Newton methods. We also include one additional method to our comparison, namely, the Gradient Method, to illustrate the difference between linearly and superlinearly convergent algorithms.

We would like to stress that the main goal of our experiments is to confirm theory and get a general idea about the actual relation between greedy and classical quasi-Newton methods in practice. There is no goal to perform exhaustive numerical testing involving many different methods and problems.

4.5.1 Regularized Log-Sum-Exp

First, let us consider the following test function:

$$f(x) \coloneqq \ln\Big(\sum_{j=1}^{m} \exp(\langle c_j, x\rangle - b_j)\Big) + \frac{1}{2}\sum_{j=1}^{m}\langle c_j, x\rangle^2 + \frac{\gamma}{2}\|x\|^2, \tag{4.5.1}$$

where $x \in \mathbb{R}^n$, $c_1, \dots, c_m \in \mathbb{R}^n$, $b_1, \dots, b_m \in \mathbb{R}$, $\gamma > 0$, and $m \geq n$.

We compare Algorithm 4.3.1 (implementing GrDFP, GrBFGS and GrSR1, depending on the choice of χ_k) with the usual Gradient Method (GM)[5] and standard quasi-Newton methods DFP, BFGS and SR1.

[5]For GM, we use the constant step size $1/L$, where L is the estimate of the Lipschitz constant of the gradient given by (4.5.5).

All the standard methods need access only to the gradient of f:

$$\nabla f(x) = g(x) + \sum_{j=1}^{m} \langle c_j, x \rangle c_j + \gamma x, \qquad g(x) \coloneqq \sum_{j=1}^{m} \pi_j(x) c_j, \tag{4.5.2}$$

where

$$\pi_j(x) \coloneqq \frac{\exp(\langle c_j, x \rangle - b_j)}{\sum_{j'=1}^{m} \exp(\langle c_{j'}, x \rangle - b_{j'})} \in [0, 1], \qquad j = 1, \ldots, m.$$

Note that, for a given point $x \in \mathbb{R}^n$, $\nabla f(x)$ can be computed in $O(mn)$ operations.

For the greedy methods, to implement the Hessian approximation update, at every iteration, we need to carry out some additional operations with the Hessian

$$\begin{aligned} \nabla^2 f(x) &= \sum_{j=1}^{m} \pi_j(x) c_j c_j^T - g(x) g(x)^T + \sum_{j=1}^{m} c_j c_j^T + \gamma I \\ &= \sum_{j=1}^{m} \big(\pi_j(x) + 1\big) c_j c_j^T - g(x) g(x)^T + \gamma I. \end{aligned} \tag{4.5.3}$$

Namely, given a point $x \in \mathbb{R}^n$, we need to be able to perform the following two actions:

- For *all* $1 \le i \le n$, compute the values

$$\langle \nabla^2 f(x) e_i, e_i \rangle = \sum_{j=1}^{m} \big(\pi_j(x) + 1\big) \langle c_j, e_i \rangle^2 - \langle g(x), e_i \rangle^2 + \gamma,$$

  where $e_1, \ldots, e_n$ are the basis vectors.

- For a given direction $h \in \mathbb{R}^n$, compute the Hessian-vector product

$$\nabla^2 f(x) h = \sum_{j=1}^{m} \big(\pi_j(x) + 1\big) \langle c_j, h \rangle c_j - \langle g(x), h \rangle g(x) + \gamma h.$$

Let us choose the standard basis in $\mathbb{R}^n$:

$$e_i \coloneqq (0, \ldots, 0, 1, 0, \ldots, 0)^T, \qquad 1 \le i \le n. \tag{4.5.4}$$

Then, both the above operations have a cost of $O(mn)$.

In particular, we see that the *cost of one iteration is comparable for all methods under our consideration.*

Note that, for our basis (4.5.4), the matrix B, defined by (4.1.11), is the identity matrix:

$$B = I.$$

Hence, the Lipschitz constant of the gradient of f w.r.t. B can be set in the following way (see (4.5.3)):

$$L = 2\sum_{j=1}^{m} \|c_j\|^2 + \gamma. \tag{4.5.5}$$

All quasi-Newton methods in our comparison start from the same initial Hessian approximation $G_0 = LB$, and use unit step sizes.

Finally, for greedy quasi-Newton methods, we also need to provide an estimate of the parameter of strong self-concordance. Note that the function f is 1-strongly convex and its Hessian is 2-Lipschitz continuous[6] w.r.t. the operator $\sum_{j=1}^{m} c_j c_j^T$ (see, e.g., [54, Ex. 1]). Hence, in view of Lemma 3.3.2, the parameter of strong self-concordance can be chosen as follows:

$$M = 2.$$

The data defining the test function (4.5.1) is randomly generated in the following way. First, we generate a collection of random vectors

$$\hat{c}_1, \ldots, \hat{c}_m$$

with entries uniformly distributed in the interval $[-1, 1]$. Then we generate $b_1, \ldots, b_m$ from the same distribution. Using this data, we form a preliminary function

$$\hat{f}(x) \coloneqq \ln\Big(\sum_{j=1}^{m} \exp(\langle \hat{c}_j, x\rangle - b_j)\Big),$$

and finally define

$$c_j \coloneqq \hat{c}_j - \nabla \hat{f}(0), \qquad j = 1, \ldots, m.$$

[6] Note, however, that the "standard" Lipschitz constant L_2 of the Hessian of f (measured w.r.t. the standard Euclidean norm in $\mathbb{R}^n$) may be significantly bigger than 2 depending on the relation between vectors $c_1, \ldots, c_m$.

Note that by construction, according to (4.5.2),

$$\nabla f(0) = \frac{1}{\sum_{j=1}^m \exp(-b_j)} \sum_{j=1}^m \exp(-b_j)\big(\hat{c}_j - \nabla \hat{f}(0)\big) = 0,$$

so the unique minimizer of our test function (4.5.1) is $x^* = 0$. The starting point x_0 for all methods is the same and generated randomly from the uniform distribution on the standard Euclidean sphere of radius $1/n$ (this choice is motivated by (4.3.24)) centered at the minimizer.

Thus, our test function (4.5.1) has three parameters: the dimension n, the number m of linear functions, and the regularization coefficient γ. Let us present computational results for different values of these parameters. The termination criterion for all methods is $f(x_k) - f(x^*) \leq \varepsilon\big(f(x_0) - f(x^*)\big)$.

In the tables below, for each method, we display the number of iterations until its termination. The minus sign $(-)$ means that the method has not been able to achieve the required accuracy after $1000n$ iterations.

Table 4.5.1: $n = m = 50$, $\gamma = 1$

ε	GM	DFP	BFGS	SR1	GrDFP	GrBFGS	GrSR1
10^{-1}	79	4	4	3	45	35	34
10^{-3}	1812	777	57	18	342	57	52
10^{-5}	5263	1866	107	29	738	72	58
10^{-7}	8873	2836	158	39	917	83	63
10^{-9}	12532	3911	203	48	1028	93	67

Table 4.5.2: $n = m = 50$, $\gamma = 0.1$

ε	GM	DFP	BFGS	SR1	GrDFP	GrBFGS	GrSR1
10^{-1}	76	4	4	3	44	33	33
10^{-3}	2732	1278	78	23	512	70	56
10^{-5}	29785	12923	254	57	3850	126	72
10^{-7}	–	23245	346	74	6794	169	81
10^{-9}	–	32441	381	79	8216	204	87

Table 4.5.3: $n = m = 250$, $\gamma = 1$

ε	GM	DFP	BFGS	SR1	GrDFP	GrBFGS	GrSR1
10^{-1}	444	4	4	3	214	158	157
10^{-3}	10351	4743	98	21	3321	264	251
10^{-5}	73685	31468	288	55	15637	350	274
10^{-7}	159391	58138	450	82	21953	413	296
10^{-9}	249492	85218	627	110	25500	464	314

Table 4.5.4: $n = m = 250$, $\gamma = 0.1$

ε	GM	DFP	BFGS	SR1	GrDFP	GrBFGS	GrSR1
10^{-1}	442	4	4	3	209	155	155
10^{-3}	9312	4175	91	21	2686	258	251
10^{-5}	207978	102972	488	87	60461	556	346
10^{-7}	–	–	1003	170	147076	792	391
10^{-9}	–	–	1407	233	212100	976	419

We see that all quasi-Newton methods outperform the Gradient Method and demonstrate superlinear convergence (from some moment, the difference in the number of iterations between successive rows in the table becomes smaller and smaller). Among quasi-Newton methods (both the standard and the greedy ones), SR1 is always better than BFGS, while DFP is significantly worse than the other two. At the first few iterations, the greedy methods lose to the standard ones, but later they catch up. However, the classical SR1 Method always remains the best. Nevertheless, the greedy methods are quite competitive.

Now let us look at the quality of Hessian approximations, produced by the quasi-Newton methods. In the tables below, we display the desired accuracy ε vs the final Hessian approximation error (defined as the operator norm of $G_k - \nabla^2 f(x_k)$ measured w.r.t. $\nabla^2 f(x_k)$). We look at the same problems as in Tables 4.5.1 and 4.5.3.

Table 4.5.5: $n = m = 50$, $\gamma = 1$

ε	DFP	BFGS	SR1	GrDFP	GrBFGS	GrSR1
10^{-0}	$1.6 \cdot 10^3$	$1.6 \cdot 10^3$	$1.6 \cdot 10^3$	$1.6 \cdot 10^3$	$1.6 \cdot 10^3$	$1.6 \cdot 10^3$
10^{-1}	$1.6 \cdot 10^3$	$1.6 \cdot 10^3$	$1.6 \cdot 10^3$	$2.7 \cdot 10^3$	$1.5 \cdot 10^3$	$1.5 \cdot 10^3$
10^{-3}	$1.6 \cdot 10^3$	$1.6 \cdot 10^3$	$1.6 \cdot 10^3$	$1.2 \cdot 10^3$	$1.2 \cdot 10^1$	$3.8 \cdot 10^0$
10^{-5}	$1.6 \cdot 10^3$	$1.6 \cdot 10^3$	$1.6 \cdot 10^3$	$2.1 \cdot 10^2$	$7.2 \cdot 10^0$	$2.6 \cdot 10^0$
10^{-7}	$1.6 \cdot 10^3$	$1.6 \cdot 10^3$	$1.6 \cdot 10^3$	$9.1 \cdot 10^1$	$5.6 \cdot 10^0$	$2.2 \cdot 10^0$
10^{-9}	$1.6 \cdot 10^3$	$1.6 \cdot 10^3$	$1.6 \cdot 10^3$	$5.2 \cdot 10^1$	$4.1 \cdot 10^0$	$1.8 \cdot 10^0$

Table 4.5.6: $n = m = 250$, $\gamma = 1$

ε	DFP	BFGS	SR1	GrDFP	GrBFGS	GrSR1
10^{-0}	$4.1 \cdot 10^4$	$4.1 \cdot 10^4$	$4.1 \cdot 10^4$	$4.1 \cdot 10^4$	$4.1 \cdot 10^4$	$4.1 \cdot 10^4$
10^{-1}	$4.1 \cdot 10^4$	$4.1 \cdot 10^4$	$4.1 \cdot 10^4$	$7.1 \cdot 10^4$	$3.8 \cdot 10^4$	$3.9 \cdot 10^4$
10^{-3}	$4.1 \cdot 10^4$	$4.1 \cdot 10^4$	$4.1 \cdot 10^4$	$6.8 \cdot 10^4$	$6.6 \cdot 10^1$	$1.7 \cdot 10^1$
10^{-5}	$4.1 \cdot 10^4$	$4.1 \cdot 10^4$	$4.1 \cdot 10^4$	$9.4 \cdot 10^3$	$3.7 \cdot 10^1$	$1.2 \cdot 10^1$
10^{-7}	$4.1 \cdot 10^4$	$4.1 \cdot 10^4$	$4.1 \cdot 10^4$	$3.1 \cdot 10^3$	$2.8 \cdot 10^1$	$9.7 \cdot 10^0$
10^{-9}	$4.1 \cdot 10^4$	$4.1 \cdot 10^4$	$4.1 \cdot 10^4$	$1.7 \cdot 10^3$	$2.2 \cdot 10^1$	$7.3 \cdot 10^0$

As we can see from these tables, for standard quasi-Newton methods the Hessian approximation error always stays at the initial level. In contrast, for the greedy ones, it decreases relatively fast (especially for GrBFGS and GrSR1). Note also that sometimes the initial residual slightly increases at the first several iterations (which is noticeable only for GrDFP). This happens due to the fact that the objective function is non-quadratic, and we apply the correction strategy.

Note that in all the above tests we have used the same values for the parameters n and m. Let us briefly illustrate what happens when $m > n$.

Table 4.5.7: $n = 50$, $m = 100$, $\gamma = 0.1$

ε	GM	DFP	BFGS	SR1	GrDFP	GrBFGS	GrSR1
10^{-1}	84	4	4	3	46	37	37
10^{-3}	897	316	32	11	183	53	52
10^{-5}	2421	833	67	19	334	63	58
10^{-7}	4087	1304	98	25	423	71	62
10^{-9}	5810	1859	132	32	473	78	66

Table 4.5.8: $n = 50$, $m = 200$, $\gamma = 0.1$

ε	GM	DFP	BFGS	SR1	GrDFP	GrBFGS	GrSR1
10^{-1}	108	4	4	3	45	46	46
10^{-3}	479	101	17	7	97	53	52
10^{-5}	1059	338	39	12	154	62	59
10^{-7}	1817	615	62	18	206	67	64
10^{-9}	2659	807	81	21	234	73	68

Comparing these tables with Table 4.5.2, we see that, with the increase of m, all the methods generally terminate faster. However, the overall picture is still the same as before.

Finally, let us present the results for the *randomized* version of Algorithm 4.3.1, in which, at every step, we select the update direction uniformly at random from the standard Euclidean sphere:

$$u_k \sim \mathrm{Unif}(\mathcal{S}^{n-1}), \tag{4.5.6}$$

where $\mathcal{S}^{n-1} \coloneqq \{x \in \mathbb{R}^n : \|x\| = 1\}$. We call the corresponding methods RaDFP, RaBFGS and RaSR1.

Table 4.5.9: $n = m = 50$, $\gamma = 1$

ε	RaDFP	RaBFGS	RaSR1
10^{-1}	35	29	34
10^{-3}	566	102	64
10^{-5}	1156	125	77
10^{-7}	1481	142	85
10^{-9}	1698	156	91

Table 4.5.10: $n = m = 250$, $\gamma = 1$

ε	RaDFP	RaBFGS	RaSR1
10^{-1}	261	144	158
10^{-3}	4276	366	287
10^{-5}	19594	517	346
10^{-7}	33293	619	376
10^{-9}	41177	698	396

It is instructive to compare these tables with Tables 4.5.1 and 4.5.3, which contain the results for the greedy methods on the same problems. We see that the randomized methods are slightly slower than the greedy

ones. However, the difference is not really significant, and, what is especially interesting, the randomized methods do not lose superlinear convergence.

4.5.2 Logistic Regression

Now let us consider another test function, namely *l_2-regularized logistic regression*, which is popular in the field of Machine Learning:

$$f(x) \coloneqq \sum_{j=1}^{m} \ln\bigl(1 + \exp(-b_j \langle c_j, x\rangle)\bigr) + \frac{\gamma}{2}\|x\|^2, \qquad x \in \mathbb{R}^n, \tag{4.5.7}$$

where $c_1, \dots, c_m \in \mathbb{R}^n$, $b_1, \dots, b_m \in \{-1, 1\}$, $\gamma > 0$, and $m \gg n$.

Note that the structure of the function (4.5.7) is similar to that of the function (4.5.1). In particular, both the diagonal of the Hessian and the Hessian-vector product for this function can be computed with similar complexity to that of computing the gradient. It can also be shown that the Lipschitz constant of the gradient of f can be chosen in according to (4.5.5) but with the coefficient $1/4$ instead of 2.

We follow the same experiment design as before with only a couple of differences. First, instead of generating the data defining the function (4.5.7) artificially, we take it from the LIBSVM collection of real-world data sets for binary classification problems[7] [32]. Second, we have found it better in practice not to apply the correction strategy in the greedy methods (i.e., simply set $\tilde{G}_k = G_k$ in Algorithm 4.3.1). This is the only heuristic that we use. For the regularization coefficient, we always use the value $\gamma = 1$, which is a standard choice.

It is not difficult to see that, for our particular problem, all the methods we consider have a comparable cost of one iteration.

Let us now look at the results.

Table 4.5.11: Data set *ijcnn1* ($n = 22$, $m = 49990$)

ε	GM	DFP	BFGS	SR1	GrDFP	GrBFGS	GrSR1
10^{-1}	246	43	8	6	25	19	18
10^{-3}	1925	672	45	16	71	25	23
10^{-5}	5123	2007	85	25	145	32	23
10^{-7}	8966	2738	102	29	192	38	23
10^{-9}	12815	3269	118	33	215	43	24

[7] The original labels b_i in the *mushrooms* data set are "1" and "2" instead of "1" and "−1". Therefore, we renamed in advance the class label "2" into "−1".

Table 4.5.12: Data set *mushrooms* ($n = 112$, $m = 8124$)

ε	GM	DFP	BFGS	SR1	GrDFP	GrBFGS	GrSR1
10^{-1}	4644	936	15	6	230	83	82
10^{-3}	77103	30594	105	24	1185	149	113
10^{-5}	–	58221	166	34	1700	170	113
10^{-7}	–	83740	217	42	1945	182	113
10^{-9}	–	107471	257	48	2088	194	114

Table 4.5.13: Data set *a9a* ($n = 123$, $m = 32561$)

ε	GM	DFP	BFGS	SR1	GrDFP	GrBFGS	GrSR1
10^{-1}	160	32	10	6	110	81	81
10^{-3}	18690	9229	145	38	2203	127	117
10^{-5}	–	79014	411	88	23715	316	123
10^{-7}	–	–	553	113	35700	441	124
10^{-9}	–	–	581	118	38285	475	124

Table 4.5.14: Data set *w8a* ($n = 300$, $m = 49749$)

ε	GM	DFP	BFGS	SR1	GrDFP	GrBFGS	GrSR1
10^{-1}	10148	3531	35	10	694	300	300
10^{-3}	194813	86315	178	34	1426	307	301
10^{-5}	–	188561	300	54	1849	327	301
10^{-7}	–	255224	387	68	2036	339	301
10^{-9}	–	264346	399	69	2057	340	301

As we can see, the general picture is the same as for the previous test function. In particular, the DFP update is always much worse than BFGS and SR1. The greedy methods are competitive with the standard ones and often outperform them for high values of accuracy.

4.6 Discussion

We have presented a new class of *greedy* quasi-Newton methods. They are based on standard quasi-Newton update formulas from the Broyden family but use greedily selected basis vectors instead of search directions for updating Hessian approximations.

Compared to classical quasi-Newton methods, greedy methods additionally display linear convergence of Hessian approximations to the true Hes-

sian. Furthermore, the rate of superlinear convergence for the iterates of the greedy methods is asymptotically faster than that of the classical methods.

However, these advantages come at a price. Namely, at every iteration of the greedy methods, one needs to compute a basis vector, which maximizes a certain measure of progress. This requires additional information beyond just the gradient of the objective function, such as the diagonal of the Hessian. If the objective function does not possess any specific structure (separable, sparse, etc.), the corresponding computations may be quite expensive.

In this regard, a natural idea is to replace the greedy strategy with a *randomized* one, where the update direction is chosen from some distribution which is easy to sample from (e.g., the uniform distribution on the unit sphere). For such algorithms, it becomes possible to prove, in the probabilistic sense, similar efficiency estimates to those of the greedy methods (see [110]). This is confirmed by our experiments, in which the corresponding scheme (Algorithm 4.3.1 with directions (4.5.6)) demonstrates almost the same performance as the greedy one.

Finally, observe that, for the greedy methods, in contrast to the classical ones, we have obtained exactly the same efficiency estimate for all updates from the extended convex Broyden class. This is a consequence of the fact that, at some point, we upper bounded all members of this class via the worst one (DFP). This was done deliberately since otherwise the greedy rule for selecting an update direction for, say, BFGS or SR1, would have been too complicated for any practical use. Nevertheless, in principle, for more sophisticated strategies for selecting update directions, we could indeed obtain much better efficiency guarantees for greedy variants of BFGS and SR1. Based on our results, it was shown recently that these improved efficiency guarantees can also be achieved using certain efficiently implementable randomized strategies (see [110]).

Chapter 5

Subgradient Ellipsoid Method

We now turn our attention to a completely different member of the quasi-Newton family of methods, namely, the *Ellipsoid Method*.

Compared to standard quasi-Newton methods for Smooth Optimization, the Ellipsoid Method does not have any local superlinear convergence. However, it has very strong *global* convergence guarantees: it exhibits a linear convergence rate with a constant depending only on the dimension of the space. Furthermore, the Ellipsoid Method is more universal in the sense that it can be readily applied to general nonsmooth problems with convex structure such as nonsmooth convex minimization problems with functional constraints, saddle-point problems, variational inequalities, etc.

In this chapter, we address one of the issues of the Ellipsoid Method, namely, its "incorrect" dependence on the *dimension* of the space.

To explain the issue, let us consider the minimization problem

$$\min_{x \in Q} f(x), \tag{5.0.1}$$

where $f \colon \mathbb{E} \to \mathbb{R}$ is a convex function, and Q is the Euclidean ball of radius $R > 0$ centered at the origin:

$$Q \coloneqq B(0, R) \equiv \{x \in \mathbb{E} : \|x\| \leq R\},$$

where $\|\cdot\| \coloneqq \|\cdot\|_B$ for some $B \in \mathcal{S}_{++}(\mathbb{E}, \mathbb{E}^*)$. The Ellipsoid Method for

solving (5.0.1) can be written as follows (see Algorithm 2.7.2):

$$\begin{aligned} x_{k+1} &:= x_k - \frac{1}{n+1} \frac{W_k g_k}{\langle g_k, W_k g_k \rangle^{1/2}}, \\ W_{k+1} &:= \frac{n^2}{n^2-1} \left(W_k - \frac{2}{n+1} \frac{W_k g_k g_k^* W_k}{\langle g_k, W_k g_k \rangle} \right), \qquad k \geq 0, \end{aligned} \tag{5.0.2}$$

where $x_0 := 0$ ($\in \mathbb{E}$), $W_0 := R^2 B^{-1}$, and $g_k := f'(x_k)$ is an arbitrary nonzero subgradient whenever $x_k \in \operatorname{int} Q$ and $g_k := Bx_k$ is a separator of x_k from Q whenever $x_k \notin \operatorname{int} Q$.

To solve problem (5.0.1) with accuracy $\varepsilon > 0$ (in terms of the function value), the Ellipsoid Method needs

$$O\left(n^2 \ln \frac{2MR}{\varepsilon} \right) \tag{5.0.3}$$

iterations, where $M > 0$ is the Lipschitz constant of f on Q (see (2.7.26)). Looking at this estimate, we can see an immediate drawback: it directly depends on the dimension and becomes useless when $n \to \infty$. In particular, we cannot guarantee any reasonable rate of convergence for the Ellipsoid Method when the dimensionality of the problem is sufficiently big.

Note that the aforementioned drawback is an artifact of the method itself, not its analysis. Indeed, when $n \to \infty$, iteration (5.0.2) reads

$$x_{k+1} := x_k, \quad W_{k+1} := W_k, \quad k \geq 0.$$

Thus, the method stays at the same point and does not make any progress.

On the other hand, the simplest Subgradient Method for solving (5.0.1) possesses the "dimension-independent"[1]

$$O\left(\frac{M^2 R^2}{\varepsilon^2} \right) \tag{5.0.4}$$

iteration complexity bound (see (2.6.13)). Comparing (5.0.3) with (5.0.4), we see that the Ellipsoid Method is significantly faster than the Subgradient Method only when n is not too big compared to MR/ε and significantly slower otherwise. Clearly, this is rather strange because the former algorithm does much more work at every iteration by "improving" the "met-

[1] Of course, complexity bound (5.0.4) may not be exactly dimension-independent, as the constants M and R may, in principle, themselves depend on the dimension n. Nevertheless, at least, there is no *explicit* dependence on n in (5.0.4) (in contrast to (5.0.3)), and there are indeed cases when both M and R are actually independent of n.

ric" W_k which is used for measuring the norm of the subgradients.

In this chapter, we propose a new ellipsoid-type algorithm for solving general nonsmooth problems with convex structure which does not have the drawback discussed above. This algorithm can be seen as a combination of the Subgradient and Ellipsoid methods and its convergence rate is basically as good as the best of the two corresponding rates (up to some logarithmic factors). In particular, when $n \to \infty$, the convergence rate of the new algorithm coincides with that of the Subgradient Method.

We would like to clarify that we are not interested in simply obtaining *any* method whose complexity is exactly the best among those of the Subgradient and Ellipsoid methods (this goal is easily achieved by running both methods in parallel). Instead, we are interested in a deeper understanding of the Ellipsoid Method and how to make it truly *continuous* in the dimension n. By properly combining two methods into one scheme, we hope to obtain a *universal* method, which will open up possibilities for further acceleration.

Contents

This chapter follows [162] (with a few additional minor clarifications regarding the choice of parameters in the general algorithmic scheme) and has the following structure.

First, in Section 5.1.1, we review the general formulation of a problem with convex structure and the associated notions of *accuracy certificate* and *residual*. Our presentation mostly follows [125] with examples taken from [130]. Then, in Section 5.1.2, we introduce the notions of *accuracy semicertificate* and *gap* and discuss their relation with those of accuracy certificate and residual.

In Section 5.2, we present the general algorithmic scheme of the new method. To measure the convergence rate of this scheme, we introduce the notion of *sliding gap* and establish some preliminary bounds on it.

In Section 5.3, we discuss different choices of parameters in the general scheme. First, we show that, by setting some parameters to zero, we obtain the standard Subgradient and Ellipsoid methods. Then we consider a couple of other less trivial choices which lead to two new algorithms. The principal of these new algorithms is the latter one, which is called the *Subgradient Ellipsoid Method*. We demonstrate that the convergence rate of this algorithm is basically as good as the best among those of the Subgradient and Ellipsoid methods.

In Section 5.4, we show that, for both new methods, it is possible to efficiently generate accuracy semicertificates whose gap is upper bounded by the sliding gap. We also compare our approach with the recently proposed technique from [125] for building accuracy certificates for the standard Ellipsoid Method.

In Section 5.5, we discuss how one can efficiently implement the general scheme of the Subgradient Ellipsoid Method and the corresponding procedure for generating accuracy semicertificates. In particular, we show that the time and memory requirements of the new scheme are the same as in the standard Ellipsoid Method.

Finally, in Section 5.6, we discuss some open questions.

5.1 Convex Problems and Accuracy Certificates

5.1.1 Description and Examples

In this chapter, we consider numerical algorithms for solving *problems with convex structure*. The main examples of such problems are convex minimization problems, convex-concave saddle-point problems, convex Nash equilibrium problems, and variational inequalities with monotone operators.

The general formulation of a problem with convex structure involves two objects:

- Solid $Q \subseteq \mathbb{E}$ (called the *feasible set*), represented by the *Separation Oracle*: given any point $x \in \mathbb{E}$, this oracle can check whether $x \in \operatorname{int} Q$, and if not, it reports a vector $g_Q(x) \in \mathbb{E}^* \setminus \{0\}$ which separates x from Q:
$$\langle g_Q(x), x - y \rangle \geq 0, \quad \forall y \in Q. \tag{5.1.1}$$
- Vector field $g\colon \operatorname{int} Q \to \mathbb{E}^*$, represented by the *First-Order Oracle*: given any point $x \in \operatorname{int} Q$, this oracle returns the vector $g(x)$.

In what follows, we only consider the problems satisfying the following condition:
$$\exists x^* \in Q\colon \quad \langle g(x), x - x^* \rangle \geq 0, \quad \forall x \in \operatorname{int} Q. \tag{5.1.2}$$

A numerical algorithm for solving a problem with convex structure starts at some point $x_0 \in \mathbb{E}$. At each step $k \geq 0$, it queries the oracles at the current *test point* x_k to obtain the new information about the problem, and

then somehow uses this new information to form the next test point x_{k+1}. Depending on whether $x_k \in \operatorname{int} Q$, the kth step of the algorithm is called *productive* or *nonproductive*.

The total information, obtained by the algorithm from the oracles after $k \geq 1$ steps, comprises its *execution protocol* which consists of:

- The test points $x_0, \ldots, x_{k-1} \in \mathbb{E}$.
- The set of productive steps $I_k := \{0 \leq i \leq k-1 : x_i \in \operatorname{int} Q\}$.
- The vectors $g_0, \ldots, g_{k-1} \in \mathbb{E}^*$ reported by the oracles: $g_i := g(x_i)$, if $i \in I_k$, and $g_i := g_Q(x_i)$, if $i \notin I_k$, $0 \leq i \leq k-1$.

An *accuracy certificate*, associated with the above execution protocol, is a nonnegative vector $\lambda := (\lambda_0, \ldots, \lambda_{k-1})$ such that $S_k(\lambda) := \sum_{i \in I_k} \lambda_i > 0$ (and, in particular, $I_k \neq \varnothing$). Given any solid Ω, containing Q, we can define the following *residual* of λ on Ω:

$$\varepsilon_k(\lambda) := \max_{x \in \Omega} \frac{1}{S_k(\lambda)} \sum_{i=0}^{k-1} \lambda_i \langle g_i, x_i - x \rangle, \tag{5.1.3}$$

which is easily computable whenever Ω is a simple set (e.g., a Euclidean ball). Note that

$$\varepsilon_k(\lambda) \geq \max_{x \in Q} \frac{1}{S_k(\lambda)} \sum_{i=0}^{k-1} \lambda_i \langle g_i, x_i - x \rangle \geq \max_{x \in Q} \frac{1}{S_k(\lambda)} \sum_{i \in I_k} \lambda_i \langle g_i, x_i - x \rangle \tag{5.1.4}$$

and, in particular, $\varepsilon_k(\lambda) \geq 0$ in view of (5.1.2).

In what follows, we will be interested in algorithms which can produce accuracy certificates $\lambda^{(k)}$ with $\varepsilon_k(\lambda^{(k)}) \to 0$ at a certain rate. This is a meaningful goal because, for all known instances of problems with convex structure, the residual $\varepsilon_k(\lambda)$ upper bounds a certain natural inaccuracy measure for the corresponding problem. Let us briefly review some standard examples (for more examples, see [125, 130] and the references therein).

Example 5.1.1 (Convex Minimization Problem). Consider the problem

$$f^* := \min_{x \in Q} f(x), \tag{5.1.5}$$

where $Q \subseteq \mathbb{E}$ is a solid and $f \colon \mathbb{E} \to \mathbb{R} \cup \{+\infty\}$ is closed convex and finite on $\operatorname{int} Q$.

The First-Order Oracle for (5.1.5) is $g(x) := f'(x)$, $x \in \operatorname{int} Q$, where $f'(x)$ is an arbitrary subgradient of f at x. Clearly, (5.1.2) holds for x^* being any solution of (5.1.5).

It is not difficult to verify that, in this example, the residual $\varepsilon_k(\lambda)$ upper bounds the functional residual: for $\hat{x}_k := \frac{1}{S_k(\lambda)} \sum_{i \in I_k} \lambda_i x_i$ or $x_k^* := \operatorname{argmin}\{f(x) : x \in X_k\}$, where $X_k := \{x_i : i \in I_k\}$, we have $f(\hat{x}_k) - f^* \leq \varepsilon_k(\lambda)$ and $f(x_k^*) - f^* \leq \varepsilon_k(\lambda)$.

Moreover, $\varepsilon_k(\lambda)$, in fact, upper bounds the primal-dual gap for a certain dual problem for (5.1.5). Indeed, let $f_*: \mathbb{E}^* \to \mathbb{R} \cup \{+\infty\}$ be the conjugate function of f. Then, we can represent (5.1.5) in the following dual form:

$$f^* = \min_{x \in Q} \max_{s \in \operatorname{dom} f_*} [\langle s, x \rangle - f_*(s)] = \max_{s \in \operatorname{dom} f_*} [-f_*(s) - \xi_Q(-s)], \qquad (5.1.6)$$

where $\operatorname{dom} f_* := \{s \in \mathbb{E}^* : f_*(s) < +\infty\}$ and $\xi_Q(-s) := \max_{x \in Q} \langle -s, x \rangle$. Denote $s_k := \frac{1}{S_k(\lambda)} \sum_{i \in I_k} \lambda_i g_i$ ($\in \operatorname{dom} f_*$). Then, using (5.1.4) and the convexity of f and f_*, we obtain

$$\begin{aligned}
\varepsilon_k(\lambda) &\geq \frac{1}{S_k(\lambda)} \sum_{i \in I_k} \lambda_i \langle g_i, x_i \rangle + \xi_Q(-s_k) \\
&= \frac{1}{S_k(\lambda)} \sum_{i \in I_k} \lambda_i [f(x_i) + f_*(g_i)] + \xi_Q(-s_k) \\
&\geq f(\hat{x}_k) + f_*(s_k) + \xi_Q(-s_k) \\
&= [f(\hat{x}_k) - f^*] + [f^* + f_*(s_k) + \xi_Q(-s_k)] \quad (\geq 0),
\end{aligned}$$

where the final inequality (in the parentheses) is due to (5.1.5) and (5.1.6). Thus, $\hat{x}_k$ and s_k are $\varepsilon_k(\lambda)$-approximate solutions (in terms of function value) to problems (5.1.5) and (5.1.6), respectively. Note that the same is true if we replace $\hat{x}_k$ with x_k^*.

Example 5.1.2 (Convex-Concave Saddle-Point Problem). Consider the following problem: Find $(u^*, v^*) \in U \times V$ such that

$$f(u^*, v) \leq f(u^*, v^*) \leq f(u, v^*), \quad \forall (u, v) \in U \times V, \qquad (5.1.7)$$

where U, V are solids in some finite-dimensional vector spaces $\mathbb{E}_u$, $\mathbb{E}_v$, respectively, and $f: U \times V \to \mathbb{R}$ is a continuous function which is *convex-concave*, i.e., $f(\cdot, v)$ is convex and $f(u, \cdot)$ is concave for any $u \in U$ and any $v \in V$.

In this example, we set $\mathbb{E} := \mathbb{E}_u \times \mathbb{E}_v$, $Q := U \times V$ and use the First-Order

Oracle

$$g(x) \coloneqq \big(f'_u(x), -f'_v(x)\big), \quad x \coloneqq (u,v) \in \operatorname{int} Q,$$

where $f'_u(x)$ is an arbitrary subgradient of $f(\cdot, v)$ at u and $f'_v(y)$ is an arbitrary supergradient of $f(u,\cdot)$ at v. Then, for any $x \coloneqq (u,v) \in \operatorname{int} Q$ and any $x' \coloneqq (u',v') \in Q$,

$$\langle g(x), x - x' \rangle = \langle f'_u(x), u - u' \rangle - \langle f'_v(x), v - v' \rangle \geq f(u,v') - f(u',v). \quad (5.1.8)$$

In particular, (5.1.2) holds for $x^* \coloneqq (u^*, v^*)$ in view of (5.1.7).

Let $\varphi\colon U \to \mathbb{R}$ and $\psi\colon V \to \mathbb{R}$ be the functions

$$\varphi(u) \coloneqq \max_{v \in V} f(u,v), \qquad \psi(v) \coloneqq \min_{u \in U} f(u,v).$$

In view of (5.1.7), we have $\psi(v) \leq f(u^*, v^*) \leq \varphi(u)$ for all $(u,v) \in U \times V$. Therefore, the difference $\varphi(u) - \psi(v)$ (called the *primal-dual gap*) is always nonnegative and can be used for measuring the quality of an approximate solution $x \coloneqq (u,v) \in Q$ to problem (5.1.7).

Denoting $\hat{x}_k \coloneqq \frac{1}{S_k(\lambda)} \sum_{i \in I_k} \lambda_i x_i =: (\hat{u}_k, \hat{v}_k)$ and using (5.1.4), we obtain

$$\begin{aligned}
\varepsilon_k(\lambda) &\geq \max_{x \in Q} \frac{1}{S_k(\lambda)} \sum_{i \in I_k} \lambda_i \langle g_i, x_i - x \rangle \\
&\geq \max_{u \in U, v \in V} \frac{1}{S_k(\lambda)} \sum_{i \in I_k} \lambda_i [f(u_i, v) - f(u, v_i)] \\
&\geq \max_{u \in U, v \in V} [f(\hat{u}_k, v) - f(u, \hat{v}_k)] = \varphi(\hat{u}_k) - \psi(\hat{v}_k) \quad (\geq 0),
\end{aligned}$$

where the second inequality is due to (5.1.8) and the next one follows from the convexity-concavity of f. Thus, the residual $\varepsilon_k(\lambda)$ upper bounds the primal-dual gap for the approximate solution $\hat{x}_k$.

Example 5.1.3 (Variational Inequality with Monotone Operator). Let $Q \subseteq \mathbb{E}$ be a solid and let $V\colon Q \to \mathbb{E}^*$ be a continuous operator which is *monotone*, i.e., $\langle V(x) - V(y), x - y \rangle \geq 0$ for all $x, y \in Q$. The goal is to solve the following (weak) *variational inequality*:

$$\text{Find } x^* \in Q\colon \quad \langle V(x), x - x^* \rangle \geq 0, \quad \forall x \in Q. \qquad (5.1.9)$$

Since V is continuous, this problem is equivalent to its strong variant: find $x^* \in Q$ such that $\langle V(x^*), x - x^* \rangle \geq 0$ for all $x \in Q$.

A standard tool for measuring the quality of an approximate solution

to (5.1.9) is the *dual gap function*, introduced in [6]:

$$f(x) \coloneqq \max_{y \in Q} \langle V(y), x - y \rangle, \qquad x \in Q.$$

It is easy to see that f is a convex nonnegative function which equals 0 exactly at the solutions of (5.1.9).

In this example, the First-Order Oracle is defined by $g(x) \coloneqq V(x)$, $x \in \operatorname{int} Q$. Denote $\hat{x}_k \coloneqq \frac{1}{S_k(\lambda)} \sum_{i \in I_k} \lambda_i x_i$. Then, using (5.1.4) and the monotonicity of V, we obtain

$$\begin{aligned} \varepsilon_k(\lambda) &\geq \max_{x \in Q} \frac{1}{S_k(\lambda)} \sum_{i \in I_k} \lambda_i \langle V(x_i), x_i - x \rangle \\ &\geq \max_{x \in Q} \frac{1}{S_k(\lambda)} \sum_{i \in I_k} \lambda_i \langle V(x), x_i - x \rangle = f(\hat{x}_k). \end{aligned}$$

Thus, $\varepsilon_k(\lambda)$ upper bounds the dual gap function for the approximate solution $\hat{x}_k$.

5.1.2 Establishing Convergence of Residual

For the algorithms considered in this chapter, instead of accuracy certificates and residuals, it turns out to be more convenient to speak about closely related notions of *accuracy semicertificates* and *gaps*, which we now introduce.

As before, let $x_0, \ldots, x_{k-1}$ be the test points, generated by the algorithm after $k \geq 1$ steps, and let $g_0, \ldots, g_{k-1}$ be the corresponding oracle outputs. An *accuracy semicertificate*, associated with this information, is a nonnegative vector $\lambda \coloneqq (\lambda_0, \ldots, \lambda_{k-1})$ such that $\Gamma_k(\lambda) \coloneqq \sum_{i=0}^{k-1} \lambda_i \|g_i\|_* > 0$. Given any solid Ω, containing Q, the *gap* of λ on Ω is defined in the following way:

$$\delta_k(\lambda) \coloneqq \max_{x \in \Omega} \frac{1}{\Gamma_k(\lambda)} \sum_{i=0}^{k-1} \lambda_i \langle g_i, x_i - x \rangle. \tag{5.1.10}$$

Comparing these definitions with those of accuracy certificate and residual, we see that the only difference between them is that now we use a different "normalizing" coefficient: $\Gamma_k(\lambda)$ instead of $S_k(\lambda)$. Also, in the definitions of semicertificate and gap, we do not make any distinction between productive and nonproductive steps. Note that $\delta_k(\lambda) \geq 0$.

Let us demonstrate that by making the gap sufficiently small, we can

make the corresponding residual sufficiently small as well. For this, we need the following standard assumption about our problem with convex structure (see, e.g., [125]).

Assumption 5.1.4. *The vector field g, reported by the First-Order Oracle, is semibounded:*

$$\langle g(x), y - x\rangle \le V, \quad \forall x \in \operatorname{int} Q,\ \forall y \in Q.$$

A classical example of a semibounded field is a bounded one: if there is $M \ge 0$, such that $\|g(x)\|_* \le M$ for all $x \in \operatorname{int} Q$, then g is semibounded with $V := MD$, where D is the diameter of Q. However, there exist other examples. For instance, if g is the subgradient field of a convex function $f\colon \mathbb{E} \to \mathbb{R} \cup \{+\infty\}$, which is finite and continuous on Q, then g is semibounded with $V := \max_Q f - \min_Q f$ (variation of f on Q); however, g is not bounded if f is not Lipschitz continuous (e.g., $f(x) := -\sqrt{x}$ on $Q := [0,1]$). Another interesting example is the subgradient field g of a ν-self-concordant barrier $f\colon \mathbb{E} \to \mathbb{R} \cup \{+\infty\}$ for the set Q; in this case, g is semibounded with $V := \nu$ (see, e.g., [133, Theorem 5.3.7]), while $f(x) \to +\infty$ at the boundary of Q.

Lemma 5.1.5. *Let λ be a semicertificate such that $\delta_k(\lambda) < r$, where r is the largest of the radii of Euclidean balls contained in Q. Then, λ is a certificate and*

$$\varepsilon_k(\lambda) \le \frac{\delta_k(\lambda)}{r - \delta_k(\lambda)} V.$$

Proof. Denote $\delta_k := \delta_k(\lambda)$, $\Gamma_k := \Gamma_k(\lambda)$, $S_k := S_k(\lambda)$. Let $\bar{x} \in Q$ be such that $B(\bar{x}, r) \subseteq Q$. For each $0 \le i \le k-1$, let z_i be a maximizer of $z \mapsto \langle g_i, z - \bar{x}\rangle$ on $B(\bar{x}, r)$. Then, for any $0 \le i \le k-1$, we have $\langle g_i, \bar{x} - x_i\rangle = \langle g_i, z_i - x_i\rangle - r\|g_i\|_*$ with $z_i \in Q$. Therefore,

$$\sum_{i=0}^{k-1} \lambda_i \langle g_i, \bar{x} - x_i\rangle = \sum_{i=0}^{k-1} \lambda_i \langle g_i, z_i - x_i\rangle - r\Gamma_k \le S_k V - r\Gamma_k, \qquad (5.1.11)$$

where the inequality follows from the separation property (5.1.1) and Assumption 5.1.4.

Let $x \in \Omega$ be arbitrary. For $y := \big(\delta_k \bar{x} + (r - \delta_k)x\big)/r \in \Omega$, we obtain

$$\begin{aligned}(r - \delta_k) \sum_{i=0}^{k-1} \lambda_i \langle g_i, x_i - x \rangle & \\ = r \sum_{i=0}^{k-1} \lambda_i \langle g_i, x_i - y \rangle + \delta_k \sum_{i=0}^{k-1} \lambda_i \langle g_i, \bar{x} - x_i \rangle & \qquad (5.1.12) \\ \le r\delta_k \Gamma_k + \delta_k \sum_{i=0}^{k-1} \lambda_i \langle g_i, \bar{x} - x_i \rangle \le \delta_k S_k V, &\end{aligned}$$

where the inequalities follow from the definition (5.1.10) of δ_k and (5.1.11), respectively.

It remains to show that λ is a certificate, i.e., $S_k > 0$. But this is simple. Indeed, if $S_k = 0$, then, taking $x := \bar{x}$ in (5.1.12) and using (5.1.11), we get $0 \ge \sum_{i=0}^{k-1} \lambda_i \langle g_i, x_i - \bar{x} \rangle \ge r\Gamma_k$, which contradicts our assumption that λ is a semicertificate, i.e., $\Gamma_k > 0$. □

According to Lemma 5.1.5, from the convergence rate of the gap $\delta_k(\lambda^{(k)})$ to zero, we can easily obtain the corresponding convergence rate of the residual $\varepsilon_k(\lambda^{(k)})$. In particular, to ensure that $\varepsilon_k(\lambda^{(k)}) \le \varepsilon$ for some $\varepsilon > 0$, it suffices to make $\delta_k(\lambda^{(k)}) \le \delta(\varepsilon) := \varepsilon r/(\varepsilon + V)$. For this reason, in the rest of this chapter, we can focus our attention on studying the convergence rate only for the gap.

5.2 General Algorithmic Scheme

Consider the general scheme presented in Algorithm 5.2.1. This scheme works with an arbitrary oracle $\mathcal{G}\colon \mathbb{E} \to \mathbb{E}^*$ satisfying the following condition:

$$\exists x^* \in B(x_0, R)\colon \quad \langle \mathcal{G}(x), x - x^* \rangle \ge 0, \quad \forall x \in \mathbb{E}. \qquad (5.2.1)$$

The point x^* from (5.2.1) is typically called a *solution* of our problem. For the general problem with convex structure, represented by the First-Order Oracle g and the Separation Oracle g_Q for the solid Q, the oracle $\mathcal{G}$ is usually defined as follows: $\mathcal{G}(x) := g(x)$, if $x \in \operatorname{int} Q$, and $\mathcal{G}(x) := g_Q(x)$, otherwise. To ensure (5.2.1), the constant R needs to be chosen sufficiently big so that $Q \subseteq B(x_0, R)$.

Algorithm 5.2.1:
General Scheme of Subgradient Ellipsoid Method

Input: Point $x_0 \in \mathbb{E}$ and scalar $R > 0$.

Initialization: Define the functions $\ell_0(x) \coloneqq 0$, $\omega_0(x) \coloneqq \frac{1}{2}\|x - x_0\|^2$.

For $k \geq 0$ **iterate:**

1. Query the oracle to obtain $g_k \coloneqq \mathcal{G}(x_k)$.
2. Compute $U_k \coloneqq \max_{x \in \Omega_k \cap L_k^-} \langle g_k, x_k - x \rangle$, where
$$\Omega_k \coloneqq \{x \in \mathbb{E} : \omega_k(x) \leq \tfrac{1}{2}R^2\}, \qquad L_k^- \coloneqq \{x \in \mathbb{E} : \ell_k(x) \leq 0\}.$$
3. Choose some coefficients $a_k, b_k \geq 0$ and update the functions
$$\begin{aligned} \ell_{k+1}(x) &\coloneqq \ell_k(x) + a_k \langle g_k, x - x_k \rangle, \\ \omega_{k+1}(x) &\coloneqq \omega_k(x) + \tfrac{1}{2} b_k (U_k - \langle g_k, x_k - x \rangle) \langle g_k, x - x_k \rangle. \end{aligned} \tag{5.2.2}$$
4. Set $x_{k+1} \coloneqq \operatorname{argmin}_{x \in \mathbb{E}} [\ell_{k+1}(x) + \omega_{k+1}(x)]$.

Note that, in Algorithm 5.2.1, ω_k are strictly convex quadratic functions and ℓ_k are affine functions. Therefore, the sets Ω_k are certain ellipsoids and L_k^- are certain halfspaces (possibly degenerate).

Let us show that Algorithm 5.2.1 is a cutting-plane scheme in which the sets $\Omega_k \cap L_k^-$ are the localizers of the solution x^*.

Lemma 5.2.1. *In Algorithm 5.2.1, for all $k \geq 0$, we have $x^* \in \Omega_k \cap L_k^-$ and $\hat{Q}_{k+1} \subseteq \Omega_{k+1} \cap L_{k+1}^-$, where $\hat{Q}_{k+1} \coloneqq \{x \in \Omega_k \cap L_k^- : \langle g_k, x - x_k \rangle \leq 0\}$.*

Proof. Let us prove the claim by induction. Clearly, $\Omega_0 = B(x_0, R)$ and $L_0^- = \mathbb{E}$, hence $\Omega_0 \cap L_0^- = B(x_0, R) \ni x^*$ by (5.2.1). Suppose we have already proved that $x^* \in \Omega_k \cap L_k^-$ for some $k \geq 0$. Combining this with (5.2.1), we obtain $x^* \in \hat{Q}_{k+1}$, so it remains to show that $\hat{Q}_{k+1} \subseteq \Omega_{k+1} \cap L_{k+1}^-$. Let $x \in \hat{Q}_{k+1}$ $(\subseteq \Omega_k \cap L_k^-)$ be arbitrary. Note that $0 \leq \langle g_k, x_k - x \rangle \leq U_k$. Hence, by (5.2.2), $\ell_{k+1}(x) \leq \ell_k(x) \leq 0$ and $\omega_{k+1}(x) \leq \omega_k(x) \leq \frac{1}{2}R^2$, which means that $x \in \Omega_{k+1} \cap L_{k+1}^-$. □

Next, let us establish an important representation of the ellipsoids Ω_k via the functions ℓ_k and the test points x_k. For this, let us define $G_k \coloneqq \nabla^2 \omega_k(0)$ for each $k \geq 0$. Observe that these operators satisfy the following simple relations (cf. (5.2.2)):

$$G_0 = B, \qquad G_{k+1} = G_k + b_k g_k g_k^*, \quad k \geq 0. \tag{5.2.3}$$

Also, let us define the sequence $R_k > 0$ by the recurrence

$$R_0 = R, \qquad R_{k+1}^2 = R_k^2 + (a_k + \tfrac{1}{2} b_k U_k)^2 \frac{(\|g_k\|_{G_k}^*)^2}{1 + b_k (\|g_k\|_{G_k}^*)^2}, \quad k \geq 0. \tag{5.2.4}$$

Lemma 5.2.2. *In Algorithm 5.2.1, for all $k \geq 0$, we have*

$$\Omega_k = \{x \in \mathbb{E} : -\ell_k(x) + \tfrac{1}{2}\|x - x_k\|_{G_k}^2 \leq \tfrac{1}{2} R_k^2\}.$$

In particular, for all $k \geq 0$ and all $x \in \Omega_k \cap L_k^-$, we have $\|x - x_k\|_{G_k} \leq R_k$.

Proof. Let $\psi_k \colon \mathbb{E} \to \mathbb{R}$ be the function $\psi_k(x) \coloneqq \ell_k(x) + \omega_k(x)$. Note that ψ_k is a quadratic function with Hessian G_k and minimizer x_k. Hence, for any $x \in \mathbb{E}$, we have

$$\psi_k(x) = \psi_k^* + \tfrac{1}{2}\|x - x_k\|_{G_k}^2, \tag{5.2.5}$$

where $\psi_k^* \coloneqq \min_{x \in \mathbb{E}} \psi_k(x)$.

Let us compute ψ_k^*. Combining (5.2.2), (5.2.5) and (5.2.3), for any $x \in \mathbb{E}$, we obtain

$$\begin{aligned} \psi_{k+1}(x) &= \psi_k(x) + (a_k + \tfrac{1}{2} b_k U_k)\langle g_k, x - x_k\rangle + \tfrac{1}{2} b_k \langle g_k, x - x_k\rangle^2 \\ &= \psi_k^* + \tfrac{1}{2}\|x - x_k\|_{G_k}^2 + (a_k + \tfrac{1}{2} b_k U_k)\langle g_k, x - x_k\rangle \\ &\quad + \tfrac{1}{2} b_k \langle g_k, x - x_k\rangle^2 \\ &= \psi_k^* + \tfrac{1}{2}\|x - x_k\|_{G_{k+1}}^2 + (a_k + \tfrac{1}{2} b_k U_k)\langle g_k, x - x_k\rangle. \end{aligned} \tag{5.2.6}$$

Therefore,

$$\begin{aligned} \psi_{k+1}^* &= \psi_k^* - \tfrac{1}{2}(a_k + \tfrac{1}{2} b_k U_k)^2 (\|g_k\|_{G_{k+1}}^*)^2 \\ &= \psi_k^* - \tfrac{1}{2}(a_k + \tfrac{1}{2} b_k U_k)^2 \frac{(\|g_k\|_{G_k}^*)^2}{1 + b_k (\|g_k\|_{G_k}^*)^2}, \end{aligned} \tag{5.2.7}$$

where the last identity follows from the fact that $G_{k+1}^{-1} g_k = G_k^{-1} g_k / (1 + b_k (\|g_k\|_{G_k}^*)^2)$ (since $G_{k+1} G_k^{-1} g_k = (1 + b_k (\|g_k\|_{G_k}^*)^2) g_k$ in view of (5.2.3)). Since (5.2.7) is true for any $k \geq 0$ and since $\psi_0^* = 0$, we thus obtain, in view of (5.2.4), that

$$\psi_k^* = \tfrac{1}{2}(R^2 - R_k^2). \tag{5.2.8}$$

Let $x \in \Omega_k$ be arbitrary. Using the definition of $\psi_k(x)$ and (5.2.8), we

obtain

$$-\ell_k(x) + \tfrac{1}{2}\|x - x_k\|^2_{G_k} = \omega_k(x) - \psi_k^* = \omega_k(x) + \tfrac{1}{2}(R_k^2 - R^2).$$

Thus, $x \in \Omega_k \iff \omega_k(x) \le \frac{1}{2}R^2 \iff -\ell_k(x) + \frac{1}{2}\|x - x_k\|^2_{G_k} \le \frac{1}{2}R_k^2$. Hence, for any $x \in \Omega_k \cap L_k^-$, we have $\ell_k(x) \le 0$, and so $\|x - x_k\|_{G_k} \le R_k$. $\square$

Lemma 5.2.2 has several consequences. First, we see that the localizers $\Omega_k \cap L_k^-$ are contained in the ellipsoids $\{x : \|x - x_k\|_{G_k} \le R_k\}$ whose centers are the test points x_k.

Second, we get an upper bound on the maximal value of the function $-\ell_k$ over the ellipsoid Ω_k: $-\ell_k(x) \le \frac{1}{2}R_k^2$ for all $x \in \Omega_k$. This observation leads us to the following definition of the *sliding gap*:

$$\Delta_k := \max_{x \in \Omega_k} \frac{1}{\Gamma_k}[-\ell_k(x)] = \max_{x \in \Omega_k} \frac{1}{\Gamma_k} \sum_{i=0}^{k-1} a_i \langle g_i, x_i - x \rangle, \quad k \ge 1, \tag{5.2.9}$$

provided that $\Gamma_k := \sum_{i=0}^{k-1} a_i \|g_i\|_* > 0$. According to our observation,

$$\Delta_k \le \frac{R_k^2}{2\Gamma_k}. \tag{5.2.10}$$

At the same time, $\Delta_k \ge 0$ in view of Lemma 5.2.1 and (5.2.1)

Comparing the definition (5.2.9) of the sliding gap Δ_k with the definition (5.1.10) of the gap $\delta_k(a^{(k)})$ for the semicertificate $a^{(k)} := (a_0, \dots, a_{k-1})$, we see that they are almost identical. The only difference between them is that the solid Ω_k, over which the maximum is taken in the definition of the sliding gap, depends on the iteration counter k. This seems to be unfortunate because we cannot guarantee that *each* Ω_k contains the feasible set Q (as required in the definition of gap) even if so does the initial solid $\Omega_0 = B(x_0, R)$. However, this problem can be dealt with. Namely, in Section 5.4, we will show that the semicertificate $a^{(k)}$ can be efficiently converted into another semicertificate $\lambda^{(k)}$ for which $\delta_k(\lambda^{(k)}) \le \Delta_k$ when taken over the initial solid $\Omega := \Omega_0$. Thus, the sliding gap Δ_k is a meaningful measure of convergence rate of Algorithm 5.2.1 and it makes sense to call the coefficients $a^{(k)}$ a *preliminary semicertificate*.

Let us now demonstrate that, for a suitable choice of the coefficients a_k and b_k in Algorithm 5.2.1, we can ensure that the sliding gap Δ_k converges to zero.

Remark 5.2.3. From now on, in order to avoid taking into account some trivial degenerate cases, it will be convenient to make the following minor technical assumption:

$$\text{In Algorithm 5.2.1, } g_k \neq 0 \text{ for all } k \geq 0.$$

Indeed, when the oracle reports $g_k = 0$ for some $k \geq 0$, it usually means that the test point x_k, at which the oracle was queried, is, in fact, an exact solution to our problem. For example, if the standard oracle for a problem with convex structure has reported $g_k = 0$, we can terminate the method and return the certificate $\lambda \coloneqq (0, \ldots, 0, 1)$ for which the residual $\varepsilon_k(\lambda) = 0$.

Let us choose the coefficients a_k and b_k in the following way:

$$a_k \coloneqq \frac{\alpha_k R + \frac{1}{2}\theta\gamma R_k}{\|g_k\|^*_{G_k}}, \qquad b_k \coloneqq \frac{\gamma}{(\|g_k\|^*_{G_k})^2}, \qquad k \geq 0, \tag{5.2.11}$$

where $\alpha_k, \gamma, \theta \geq 0$ are certain coefficients (to be specified later). We will see that two main parameters in the above parametrization are actually α_k and γ. They control the "strength" of, respectively, the "subgradient" and "ellipsoidal" components of Algorithm 5.2.1. The coefficient θ is just a certain absolute constant. Its specific choice is not particularly important but usually helps to improve absolute constants in the final convergence rate estimate.

According to (5.2.10), to estimate the convergence rate of the sliding gap, we need to estimate the rate of growth of the coefficients R_k and Γ_k from above and below, respectively. Let us do this.

Lemma 5.2.4. *In Algorithm 5.2.1 with parameters (5.2.11), for all $k \geq 0$,*

$$R_k^2 \leq [q_c(\gamma)]^k C_k R^2, \tag{5.2.12}$$

where

$$q_c(\gamma) \coloneqq 1 + \frac{c\gamma^2}{2(1+\gamma)}, \qquad c \coloneqq \tfrac{1}{2}(\tau+1)(\theta+1)^2,$$
$$C_k \coloneqq 1 + \frac{\tau+1}{\tau}\sum_{i=0}^{k-1}\alpha_i^2,$$

and $\tau > 0$ can be chosen arbitrarily. Moreover, if $\alpha_k = 0$ for all $k \geq 0$, then, $R_k^2 = [q_c(\gamma)]^k R^2$ for all $k \geq 0$ with $c \coloneqq \frac{1}{2}(\theta+1)^2$.

Proof. By the definition of U_k and Lemma 5.2.2, we have

$$U_k = \max_{x \in \Omega_k \cap L_k^-} \langle g_k, x_k - x \rangle \leq \max_{\|x - x_k\|_{G_k} \leq R_k} \langle g_k, x_k - x \rangle = R_k \|g_k\|_{G_k}^*. \quad (5.2.13)$$

At the same time, $U_k \geq 0$ in view of Lemma 5.2.1 and (5.2.1). Hence,

$$\begin{aligned}(a_k + \tfrac{1}{2} b_k U_k)^2 \frac{(\|g_k\|_{G_k}^*)^2}{1 + b_k (\|g_k\|_{G_k}^*)^2} &\leq (a_k + \tfrac{1}{2} b_k R_k \|g_k\|_{G_k}^*)^2 \frac{(\|g_k\|_{G_k}^*)^2}{1 + b_k (\|g_k\|_{G_k}^*)^2} \\ &= \frac{1}{1+\gamma} \big(\alpha_k R + \tfrac{1}{2}(\theta+1)\gamma R_k\big)^2,\end{aligned}$$

where the identity follows from (5.2.11). Combining this with (5.2.4), we obtain

$$R_{k+1}^2 \leq R_k^2 + \frac{1}{1+\gamma} \big(\alpha_k R + \tfrac{1}{2}(\theta+1)\gamma R_k\big)^2. \quad (5.2.14)$$

Note that, for any $\xi_1, \xi_2 \geq 0$ and any $\tau > 0$, we have

$$(\xi_1 + \xi_2)^2 = \xi_1^2 + 2\xi_1\xi_2 + \xi_2^2 \leq \frac{\tau+1}{\tau}\xi_1^2 + (\tau+1)\xi_2^2 = (\tau+1)\Big(\frac{1}{\tau}\xi_1^2 + \xi_2^2\Big)$$

(look at the minimum of the right-hand side in τ). Therefore, for any $\tau > 0$,

$$R_{k+1}^2 \leq R_k^2 + \frac{\tau+1}{1+\gamma}\Big(\frac{1}{\tau}\alpha_k^2 R^2 + \tfrac{1}{4}(\theta+1)^2\gamma^2 R_k^2\Big) = q R_k^2 + \beta_k R^2,$$

where $q \coloneqq q_c(\gamma) \geq 1$ and $\beta_k \coloneqq \frac{\tau+1}{\tau(1+\gamma)}\alpha_k^2$. Dividing both sides by q^{k+1}, we get

$$\frac{R_{k+1}^2}{q^{k+1}} \leq \frac{R_k^2}{q^k} + \frac{\beta_k R^2}{q^{k+1}}.$$

Since this is true for any $k \geq 0$, we thus obtain, in view of (5.2.4), that

$$\frac{R_k^2}{q^k} \leq \frac{R_0^2}{q^0} + R^2 \sum_{i=0}^{k-1} \frac{\beta_i}{q^{i+1}} = \Big(1 + \sum_{i=0}^{k-1} \frac{\beta_i}{q^{i+1}}\Big) R^2,$$

Multiplying both sides by q^k and using the fact that $\frac{\beta_i}{q^{i+1}} \leq \frac{\tau+1}{\tau}\alpha_i^2$, we come to (5.2.12).

When $\alpha_k = 0$ for all $k \geq 0$, we have $\ell_k = 0$ and $L_k^- = \mathbb{E}$ for all $k \geq 0$. Therefore, by Lemma 5.2.2, $\Omega_k = \{x : \|x - x_k\|_{G_k} \leq R_k\}$ and hence (5.2.13)

is, in fact, an equality. Consequently, (5.2.14) becomes

$$R_{k+1}^2 = R_k^2 + \frac{c\gamma^2}{2(1+\gamma)} R_k^2 = q_c(\gamma) R_k^2,$$

where $c := \frac{1}{2}(\theta+1)^2$. □

Remark 5.2.5. From the proof, one can see that C_k in Lemma 5.2.4 can be improved up to $C_k' := 1 + \frac{\tau+1}{\tau(1+\gamma)} \sum_{i=0}^{k-1} \frac{\alpha_i^2}{[q_c(\gamma)]^{i+1}}$.

Lemma 5.2.6. *In Algorithm 5.2.1 with parameters* (5.2.11)*, for all* $k \geq 1$*,*

$$\Gamma_k \geq R\Big(\sum_{i=0}^{k-1} \alpha_i + \tfrac{1}{2}\theta \sqrt{\gamma n \big[(1+\gamma)^{k/n} - 1\big]} \Big). \tag{5.2.15}$$

Proof. By the definition of Γ_k and (5.2.11), we have

$$\Gamma_k = \sum_{i=0}^{k-1} a_i \|g_i\|_* = R \sum_{i=0}^{k-1} \alpha_i \rho_i + \tfrac{1}{2}\theta\gamma \sum_{i=0}^{k-1} R_i \rho_i,$$

where $\rho_i := \|g_i\|_* / \|g_i\|_{G_i}^*$. Let us estimate each sum from below separately.

For the first sum, we can use the trivial bound $\rho_i \geq 1$, which is valid for any $i \geq 0$ (since $G_i \succeq B$ in view of (5.2.3)). This gives us

$$\sum_{i=0}^{k-1} \alpha_i \rho_i \geq \sum_{i=0}^{k-1} \alpha_i.$$

Let us estimate the second sum. According to (5.2.4), for any $i \geq 0$, we have $R_i \geq R$. Hence,

$$\sum_{i=0}^{k-1} R_i \rho_i \geq R \sum_{i=0}^{k-1} \rho_i \geq R \Big(\sum_{i=0}^{k-1} \rho_i^2 \Big)^{1/2},$$

and it remains to lower bound $\sum_{i=0}^{k-1} \rho_i^2$. By (5.2.3) and (5.2.11), $G_0 = B$ and $G_{i+1} = G_i + \gamma g_i g_i^* / (\|g_i\|_{G_i}^*)^2$ for all $i \geq 0$. Therefore,

$$\begin{aligned} \sum_{i=0}^{k-1} \rho_i^2 &= \frac{1}{\gamma} \sum_{i=0}^{k-1} (\langle B^{-1}, G_{i+1} - G_i \rangle = \frac{1}{\gamma} \langle B^{-1}, G_k - B \rangle = \frac{1}{\gamma} [\langle B^{-1}, G_k \rangle - n] \\ &\geq \frac{n}{\gamma} \big([\det(B^{-1}, G_k)]^{1/n} - 1 \big) = \frac{n}{\gamma} \big[(1+\gamma)^{k/n} - 1 \big], \end{aligned}$$

where we have applied the arithmetic-geometric mean inequality (see Propositions 2.1.3(iv) and 2.1.4(vi)). Combining the estimates we have obtained, we arrive at (5.2.15). □

5.3 Main Instances of General Scheme

Let us consider several specific choices of parameters α_k, γ and θ in (5.2.11).

5.3.1 Subgradient Method

The simplest possibility is to choose

$$\alpha_k > 0, \qquad \gamma := 0, \qquad \theta := 0.$$

In this case, $b_k \equiv 0$, so $G_k = B$ and $\omega_k(x) = \omega_0(x) = \frac{1}{2}\|x - x_0\|^2$ for all $x \in \mathbb{E}$ and all $k \geq 0$ (see (5.2.3) and (5.2.2)). Consequently, the new test points x_{k+1} in Algorithm 5.2.1 are generated according to the following rule:

$$x_{k+1} = \operatorname*{argmin}_{x \in \mathbb{E}} \Big[\sum_{i=0}^{k} a_i \langle g_i, x - x_i \rangle + \frac{1}{2}\|x - x_0\|^2 \Big], \qquad k \geq 0,$$

where $a_i = \alpha_i R / \|g_i\|_*$. Thus, Algorithm 5.2.1 is the Subgradient Method[2]:

$$x_{k+1} = x_k - a_k g_k, \qquad k \geq 0. \tag{5.3.1}$$

In this example, each ellipsoid Ω_k is simply a ball: $\Omega_k = B(x_0, R)$ for all $k \geq 0$. Hence, the sliding gap Δ_k, defined in (5.2.9), does not "slide" and coincides with the gap of the semicertificate $a := (a_0, \ldots, a_{k-1})$ on the solid $B(x_0, R)$:

$$\Delta_k = \max_{x \in B(x_0, R)} \frac{1}{\Gamma_k} \sum_{i=0}^{k-1} a_i \langle g_i, x_i - x \rangle.$$

In view of Lemmas 5.2.4 and 5.2.6, for all $k \geq 1$, we have

$$R_k^2 \leq \Big(1 + \sum_{i=0}^{k-1} \alpha_i^2\Big) R^2, \qquad \Gamma_k \geq R \sum_{i=0}^{k-1} \alpha_i$$

[2]Note that method (5.3.1) is not exactly the "standard" Subgradient Method discussed in Section 2.6, as it uses Separation Oracle instead of projection. Nevertheless, the main properties of both methods are quite similar.

(tend $\tau \to +\infty$ in Lemma 5.2.4). Substituting these estimates into (5.2.10), we obtain the following well-known convergence rate estimate for the gap in the Subgradient Method (c.f. (2.6.4)):

$$\Delta_k \leq \frac{1 + \sum_{i=0}^{k-1} \alpha_i^2}{2 \sum_{i=0}^{k-1} \alpha_i} R.$$

The standard strategies for choosing the coefficients α_i are as follows (see Section 2.6):

1. We fix in advance the number of iterations $k \geq 1$ of the method and use *constant* coefficients:

   $$\alpha_i := \frac{1}{\sqrt{k}}, \qquad 0 \leq i \leq k-1.$$

   This corresponds to the so-called *Short-Step* Subgradient Method. For this method, we have

   $$\Delta_k \leq \frac{R}{\sqrt{k}}.$$

2. Alternatively, we can use *time-varying* coefficients:

   $$\alpha_i := \frac{1}{\sqrt{i+1}}, \qquad i \geq 0.$$

   This approach does not require us to fix in advance the number of iterations k. However, the corresponding convergence rate estimate becomes slightly worse:

   $$\Delta_k \leq \frac{\ln k + 2}{2\sqrt{k}} R.$$

   (Indeed, $\sum_{i=0}^{k-1} \alpha_i^2 = \sum_{i=1}^{k} \frac{1}{i} \leq \ln k + 1$, $\sum_{i=0}^{k-1} \alpha_i = \sum_{i=1}^{k} \frac{1}{\sqrt{i}} \geq \sqrt{k}$, see Lemma 2.6.3.)

Remark 5.3.1. If we allow projections onto the feasible set, then, for the Subgradient Method with time-varying coefficients α_i, one can establish the $O(1/\sqrt{k})$ convergence rate for the "truncated" gap

$$\Delta_{k_0,k} := \max_{x \in B(x_0,R)} \frac{1}{\Gamma_{k_0,k}} \sum_{i=k_0}^{k} a_i \langle g_i, x_i - x \rangle,$$

where $\Gamma_{k_0,k} \coloneqq \sum_{i=k_0}^{k} a_i \|g_i\|_*$, $k_0 \coloneqq \lceil k/2 \rceil$ (see Theorem 2.6.5).

5.3.2 Standard Ellipsoid Method

Another extreme choice is the following one:

$$\alpha_k \coloneqq 0, \qquad \gamma > 0, \qquad \theta \coloneqq 0. \tag{5.3.2}$$

For this choice, we have $a_k = 0$ for all $k \geq 0$. Hence, $\ell_k = 0$ and $L_k^- = \mathbb{E}$ for all $k \geq 0$. Therefore, the localizers in this method are the following ellipsoids (see Lemma 5.2.2):

$$\Omega_k \cap L_k^- = \Omega_k = \{x \in \mathbb{E} : \|x - x_k\|_{G_k} \leq R_k\}, \qquad k \geq 0. \tag{5.3.3}$$

Observe that, in this example, $\Gamma_k \equiv \sum_{i=0}^{k-1} a_i \|g_i\|_* = 0$ for all $k \geq 1$, so there is no preliminary semicertificate and the sliding gap is undefined. However, we can still ensure the convergence to zero of a certain meaningful measure of optimality, namely, the *average radius* of the localizers Ω_k:

$$\operatorname{avrad} \Omega_k \coloneqq [\operatorname{vol}(\Omega_k / B_0)]^{1/n}, \qquad k \geq 0, \tag{5.3.4}$$

where $B_0 \coloneqq B(0, 1)$ is the unit ball.

Indeed, by (5.3.3) and Propositions 2.1.8 and 2.1.4(ii), we have

$$\operatorname{avrad} \Omega_k = [\det(B^{-1}, R_k^{-2} G_k)]^{-1/(2n)} = [\det(B^{-1}, G_k)]^{-1/(2n)} R_k. \tag{5.3.5}$$

Let us define the following functions for any real $c, p > 0$:

$$q_c(\gamma) \coloneqq 1 + \frac{c\gamma^2}{2(1+\gamma)}, \qquad \zeta_{p,c}(\gamma) \coloneqq \frac{[q_c(\gamma)]^p}{1+\gamma}, \qquad \gamma > 0. \tag{5.3.6}$$

According to Lemma 5.2.4, for any $k \geq 0$, we have

$$R_k^2 = [q_{1/2}(\gamma)]^k R^2. \tag{5.3.7}$$

At the same time, in view of (5.2.3) and (5.2.11), for all $k \geq 0$,

$$\det(B^{-1}, G_k) = \prod_{i=0}^{k-1} (1 + b_i(\|g_i\|_{G_i}^*)^2) = (1+\gamma)^k. \tag{5.3.8}$$

Substituting (5.3.7) and (5.3.8) into (5.3.5) and using the definition (5.3.6),

we obtain, for any $k \geq 0$,

$$\operatorname{avrad} \Omega_k = \frac{[q_{1/2}(\gamma)]^{k/2}}{(1+\gamma)^{k/(2n)}} R = [\zeta_{n,1/2}(\gamma)]^{k/(2n)} R. \tag{5.3.9}$$

Let us now choose γ which minimizes $\operatorname{avrad} \Omega_k$. For such computations, the following auxiliary result is useful (see Section 5.A for the proof).

Lemma 5.3.2. *For any $c \geq 1/2$ and any $p \geq 2$, the function $\zeta_{p,c}$, defined in (5.3.6), attains its minimum at a unique point*

$$\gamma_c(p) := \frac{2}{\sqrt{c^2p^2 - (2c-1)} + cp - 1} \in \left[\frac{1}{cp}, \frac{2}{cp}\right] \tag{5.3.10}$$

with the corresponding optimal value

$$\zeta_{p,c}\big(\gamma_c(p)\big) \leq \exp\big(-1/(2cp)\big).$$

Applying Lemma 5.3.2 to (5.3.9), we see that the optimal value of γ is

$$\gamma := \gamma_{1/2}(n) = \frac{2}{n/2 + n/2 - 1} = \frac{2}{n-1}, \tag{5.3.11}$$

for which $\zeta_{n,1/2}(\gamma) \leq \exp(-1/n)$. With this choice of γ, we obtain, for all $k \geq 0$, that

$$\operatorname{avrad} \Omega_k \leq \exp\big(-k/(2n^2)\big) R. \tag{5.3.12}$$

One can check that Algorithm 5.2.1 with parameters (5.2.11), (5.3.2) and (5.3.11) is, in fact, the standard Ellipsoid Method (see Remark 5.5.1).

5.3.3 Ellipsoid Method with Preliminary Semicertificate

As we have seen, we cannot measure the convergence rate of the standard Ellipsoid Method using the sliding gap because there is no preliminary semicertificate in this method. Let us present a modification of the standard Ellipsoid Method which does not have this drawback but still enjoys the same convergence rate as the original method (up to absolute constants).

For this, let us choose the parameters in the following way:

$$\alpha_k := 0, \qquad \gamma > 0, \qquad \theta := \sqrt{2} - 1 \ (\approx 0.41). \tag{5.3.13}$$

The main difference compared to (5.3.2) is that now we use a non-zero θ. The specific value suggested in (5.3.13) is not especially important and is simply motivated by the desire to decrease absolute constants in the final efficiency estimate as much as possible.

In view of Lemma 5.2.4, for all $k \geq 0$, we have

$$R_k^2 = [q_1(\gamma)]^k R^2. \tag{5.3.14}$$

Also, by Lemma 5.2.6, for all $k \geq 1$,

$$\Gamma_k \geq \frac{1}{2}\theta R \sqrt{\gamma n [(1+\gamma)^{k/n} - 1]}.$$

Thus, according to (5.2.10), for each $k \geq 1$, we obtain the following estimate for the sliding gap:

$$\Delta_k \leq \frac{[q_1(\gamma)]^k R}{\theta \sqrt{\gamma n [(1+\gamma)^{k/n} - 1]}} = \frac{1}{\theta \eta_k(\gamma, n)} [\zeta_{2n,1}(\gamma)]^{k/(2n)} R, \tag{5.3.15}$$

where

$$\eta_k(\gamma, n) \coloneqq \sqrt{\gamma n (1 - (1+\gamma)^{-k/n})} \quad (> 0),$$

and $\zeta_{2n,1}(\gamma)$ is defined in (5.3.6).

Note that the main factor in estimate (5.3.15) is $[\zeta_{2n,1}(\gamma)]^{k/(2n)}$. Let us choose γ by minimizing this expression. Applying Lemma 5.3.2, we obtain

$$\gamma \coloneqq \gamma_1(2n) \in \left[\frac{1}{2n}, \frac{1}{n}\right]. \tag{5.3.16}$$

Theorem 5.3.3. *In Algorithm 5.2.1 with parameters (5.2.11), (5.3.13) and (5.3.16), for all $k \geq 1$, we have*

$$\Delta_k \leq 6 \exp\bigl(-k/(8n^2)\bigr) R.$$

Proof. i. Suppose $k \geq n^2$. By Lemma 5.3.2,

$$\zeta_{2n,1}(\gamma) \leq \exp\bigl(-1/(4n)\bigr).$$

Hence, by (5.3.15),

$$\Delta_k \leq \frac{1}{\theta \eta_k(\gamma, n)} \exp\bigl(-k/(8n^2)\bigr) R.$$

It remains to estimate from below $\theta\eta_k(\gamma, n)$.

Since $k \geq n^2$, we have

$$(1+\gamma)^{k/n} \geq (1+\gamma)^n \geq 1+\gamma n.$$

Hence,

$$\eta_k(\gamma, n) \geq \frac{\gamma n}{\sqrt{1+\gamma n}}.$$

Note that the function $\tau \mapsto \tau/\sqrt{1+\tau}$ is increasing on $\mathbb{R}_+$. Therefore, using (5.3.16), we obtain

$$\eta_k(\gamma, n) \geq \frac{1/2}{\sqrt{1+1/2}} = \frac{1}{\sqrt{6}}.$$

Thus, for our choice of θ,

$$\theta\eta_k(\gamma, n) \geq \frac{\sqrt{2}-1}{\sqrt{6}} \geq \frac{1}{6}.$$

ii. Now suppose $k \leq n^2$. Then,

$$6\exp\bigl(-k/(8n^2)\bigr) \geq 6\exp(-1/8) \geq 5.$$

Therefore, it suffices to prove that $\Delta_k \leq 5R$ or, in view of (5.2.9), that

$$\langle g_i, x_i - x\rangle \leq 5R\|g_i\|_*,$$

where $x \in \Omega_k \cap L_k^-$ and $0 \leq i \leq k-1$ are arbitrary. Note that

$$\langle g_i, x_i - x\rangle \leq \|g_i\|_{G_i}^* \|x_i - x\|_{G_i} \leq \|g_i\|_* \|x_i - x\|_{G_i}$$

since $G_i \succeq B$ (see (5.2.3)). Hence, it remains to prove that

$$\|x_i - x\|_{G_i} \leq 5R.$$

Recall from (5.2.3) and (5.2.4) that $G_i \preceq G_k$ and $R_i \leq R_k$. Therefore,

$$\begin{aligned}
\|x_i - x\|_{G_i} &\leq \|x_i - x^*\|_{G_i} + \|x^* - x\|_{G_i} \\
&\leq \|x_i - x^*\|_{G_i} + \|x^* - x\|_{G_k} \\
&\leq \|x_i - x^*\|_{G_i} + \|x_k - x^*\|_{G_k} + \|x_k - x\|_{G_k} \\
&\leq R_i + 2R_k \leq 3R_k,
\end{aligned}$$

where the penultimate inequality follows from Lemmas 5.2.1 and 5.2.2. According to (5.3.14),

$$R_k = [q_1(\gamma)]^{k/2} R \leq [q_1(\gamma)]^{n^2/2} R$$

(recall that $q_1(\gamma) \geq 1$). Thus, it remains to show that

$$3[q_1(\gamma)]^{n^2/2} \leq 5.$$

But this is immediate. Indeed, by (5.3.6) and (5.3.16), we have

$$[q_1(\gamma)]^{n^2/2} \leq \exp\bigl(n^2\gamma^2/(4(1+\gamma))\bigr) \leq \exp(1/4),$$

so

$$3[q_1(\gamma)]^{n^2/2} \leq 3\exp(1/4) \leq 5. \qquad \square$$

5.3.4 Subgradient Ellipsoid Method

The previous algorithm still shares the drawback of the original Ellipsoid Method, namely, it does not work when $n \to \infty$. To eliminate this drawback, we will choose α_k similarly to how this is done in the Subgradient Method.

Consider the following choice of parameters:

$$\begin{gathered} \alpha_k := \sqrt{\theta/(\theta+1)}\,\beta_k, \qquad \gamma := \gamma_1(2n) \in \left[\frac{1}{2n}, \frac{1}{n}\right], \\ \theta := \sqrt[3]{2} - 1 \ (\approx 0.26), \end{gathered} \tag{5.3.17}$$

where $\gamma_1(2n)$ is defined in (5.3.10), and $\beta_k > 0$ is a new sequence of coefficients (to be specified later). A "special" coefficient $\sqrt{\theta/(\theta+1)}$, linking the old and new parameters, and a particular value of θ suggested in (5.3.17) have been chosen in such a way so as to obtain "nice" absolute constants in the following main result.

Theorem 5.3.4. *In Algorithm 5.2.1 with parameters* (5.2.11) *and* (5.3.17)*, where $\beta_0 \geq 1$, we have, for all $k \geq 1$,*

$$\Delta_k \leq \begin{cases} 2\bigl(\sum_{i=0}^{k-1} \beta_i\bigr)^{-1}\bigl(1 + \sum_{i=0}^{k-1} \beta_i^2\bigr)R, & \text{if } k \leq n^2, \\ 6\exp\bigl(-k/(8n^2)\bigr)\bigl(1 + \sum_{i=0}^{k-1} \beta_i^2\bigr)R, & \text{if } k \geq n^2. \end{cases} \tag{5.3.18}$$

Proof. Applying Lemma 5.2.4 with $\tau := \theta$ and using (5.3.17), we obtain

$$R_k^2 \le [q_1(\gamma)]^k C_k R^2, \qquad C_k = 1 + \sum_{i=0}^{k-1} \beta_i^2. \tag{5.3.19}$$

At the same time, by Lemma 5.2.6, we have

$$\Gamma_k \ge R\Big(\sqrt{\frac{\theta}{\theta+1}} \sum_{i=0}^{k-1} \beta_i + \tfrac{1}{2}\theta\sqrt{\gamma n[(1+\gamma)^{k/n} - 1]}\Big). \tag{5.3.20}$$

Note that $\frac{1}{2}\theta\sqrt{\gamma n} \le \frac{1}{2}\theta \le \sqrt{\theta/(\theta+1)}$ by (5.3.17). Since $\beta_0 \ge 1$, we thus obtain

$$\begin{aligned}\Gamma_k &\ge \tfrac{1}{2}R\theta\sqrt{\gamma n}\left(1 + \sqrt{(1+\gamma)^{k/n} - 1}\right) \ge \tfrac{1}{2}R\theta\sqrt{\gamma n}\,(1+\gamma)^{k/(2n)} \\ &\ge \tfrac{1}{2\sqrt{2}}R\theta(1+\gamma)^{k/(2n)} \ge \tfrac{1}{12}R(1+\gamma)^{k/(2n)},\end{aligned} \tag{5.3.21}$$

where the last two inequalities follow from (5.3.17). Therefore, by (5.2.10), (5.3.19) and (5.3.21),

$$\Delta_k \le \frac{R_k^2}{2\Gamma_k} \le 6\frac{[q_1(\gamma)]^k}{(1+\gamma)^{k/(2n)}} C_k R = 6[\zeta_{2n,1}(\gamma)]^{k/(2n)} C_k R,$$

where $\zeta_{2n,1}(\gamma)$ is defined in (5.3.6). Observe that, for our choice of γ, by Lemma 5.3.2, we have $\zeta_{2n,1}(\gamma) \le \exp\big(-1/(4n)\big)$. This proves the second estimate[3] in (5.3.18).

On the other hand, dropping the second term in (5.3.20), we can write

$$\Gamma_k \ge R\sqrt{\frac{\theta}{\theta+1}} \sum_{i=0}^{k-1} \beta_i. \tag{5.3.22}$$

Suppose $k \le n^2$. Then, from (5.3.6) and (5.3.17), it follows that

$$[q_1(\gamma)]^k \le [q_1(\gamma)]^{n^2} \le \exp\Big(\frac{\gamma^2 n^2}{2(1+\gamma)}\Big) \le \sqrt{e}.$$

Hence, by (5.3.19), $R_k \le \sqrt{e}\, C_k R^2$. Combining this with (5.2.10) and (5.3.22),

[3] In fact, we have proved the second estimate in (5.3.18) for all $k \ge 1$ (not only for $k \ge n^2$).

we obtain

$$\Delta_k \leq \frac{1}{2}\sqrt{\frac{e(\theta+1)}{\theta}}\Big(\sum_{i=0}^{k-1}\beta_i\Big)^{-1} C_k R.$$

By numerical evaluation, one can verify that, for our choice of θ, we have $\frac{1}{2}\sqrt{e(\theta+1)/\theta} \leq 2$. This proves the first estimate in (5.3.18). □

Exactly as in the Subgradient Method, we can use the following two strategies for choosing the coefficients β_k:

1. We fix in advance the number of iterations $k \geq 1$ of the method and use constant coefficients:

$$\beta_i := \frac{1}{\sqrt{k}}, \qquad 0 \leq i \leq k-1.$$

In this case,

$$\Delta_k \leq \begin{cases} 4R/\sqrt{k} & \text{if } k \leq n^2, \\ 12R\exp\bigl(-k/(8n^2)\bigr) & \text{if } k \geq n^2. \end{cases} \tag{5.3.23}$$

2. We use time-varying coefficients:

$$\beta_i := \frac{1}{\sqrt{i+1}}, \qquad i \geq 0.$$

In this case,

$$\Delta_k \leq \begin{cases} 2(\ln k + 2)R/\sqrt{k} & \text{if } k \leq n^2, \\ 6(\ln k + 2)R\exp\bigl(-k/(8n^2)\bigr) & \text{if } k \geq n^2. \end{cases}$$

Let us discuss convergence rate estimate (5.3.23). Up to absolute constants, this estimate is exactly the same as in the Subgradient Method when $k \leq n^2$ and as in the Ellipsoid Method when $k \geq n^2$. In particular, when $n \to \infty$, we recover the convergence rate of the Subgradient Method.

To provide a better interpretation of the obtained results, let us compare the convergence rates of the Subgradient and Ellipsoid methods:

$$\begin{aligned} \text{Subgradient Method:} &\quad 1/\sqrt{k} \\ \text{Ellipsoid Method:} &\quad \exp\bigl(-k/(2n^2)\bigr). \end{aligned}$$

To compare these rates, let us look at their squared ratio:

$$\rho_k := \left(\frac{1/\sqrt{k}}{\exp\bigl(-k/(2n^2)\bigr)} \right)^2 = \frac{1}{k} \exp(k/n^2).$$

Let us find out for which values of k the rate of the Subgradient Method is better than that of the Ellipsoid Method and vice versa. We assume that $n \geq 2$.

Note that the function $\tau \mapsto \exp(\tau)/\tau$ is strictly decreasing on $(0, 1]$ and strictly increasing on $[1, +\infty)$ (indeed, its derivative is $\exp(\tau)(\tau - 1)/\tau^2$). Hence, ρ_k is strictly decreasing in k for $1 \leq k \leq n^2$ and strictly increasing in k for $k \geq n^2$. Since $n \geq 2$, we have

$$\rho_2 = \tfrac{1}{2} \exp(2/n^2) \leq \tfrac{1}{2}\sqrt{e} \leq 1.$$

At the same time, $\rho_k \to +\infty$ when $k \to \infty$. Therefore, there exists a unique integer $K_0 \geq 2$ such that $\rho_k \leq 1$ for all $k \leq K_0$ and $\rho_k \geq 1$ for all $k \geq K_0$.

Let us estimate K_0. Clearly, for any $n^2 \leq k \leq n^2 \ln(2n)$, we have

$$\rho_k \leq \frac{\exp\bigl(n^2 \ln(2n)/n^2\bigr)}{n^2 \ln(2n)} = \frac{2}{n \ln(2n)} \leq 1,$$

while, for any $k \geq 3n^2 \ln(2n)$, we have

$$\rho_k \geq \frac{\exp\bigl(3n^2 \ln(2n)/n^2\bigr)}{3n^2 \ln(2n)} = \frac{(2n)^3}{3n^2 \ln(2n)} = \frac{8n}{3 \ln(2n)} \geq 1.$$

Hence,

$$n^2 \ln(2n) \leq K_0 \leq 3n^2 \ln(2n).$$

Thus, up to an absolute constant, $n^2 \ln(2n)$ is the switching moment, starting from which the rate of the Ellipsoid Method becomes better than that of the Subgradient Method.

Returning to the obtained convergence rate estimate (5.3.23), we see that, ignoring absolute constants and the "small" interval of the values of k between n^2 and $n^2 \ln n$, our convergence rate is basically the best among the corresponding rates of the Subgradient and Ellipsoid methods.

5.4 Constructing Accuracy Semicertificate

Let us show how to convert a preliminary accuracy semicertificate, produced by Algorithm 5.2.1, into a semicertificate whose gap on the initial solid is upper bounded by the sliding gap. The key ingredient here is the following auxiliary algorithm which was first proposed in [125] for building accuracy certificates in the standard Ellipsoid Method.

5.4.1 Augmentation Algorithm

Let $k \geq 0$ be an integer and let $Q_0, \ldots, Q_k$ be solids in $\mathbb{E}$ such that

$$\hat{Q}_i := \{x \in Q_i : \langle g_i, x - x_i \rangle \leq 0\} \subseteq Q_{i+1}, \qquad 0 \leq i \leq k-1, \tag{5.4.1}$$

where $x_i \in \mathbb{E}$, $g_i \in \mathbb{E}^*$. Further, suppose that, for any $s \in \mathbb{E}^*$ and any $0 \leq i \leq k-1$, we can compute a *dual multiplier* $\mu \geq 0$ such that

$$\max_{x \in \hat{Q}_i} \langle s, x \rangle = \max_{x \in Q_i} [\langle s, x \rangle + \mu \langle g_i, x_i - x \rangle] \tag{5.4.2}$$

(provided that certain regularity conditions hold). Let us abbreviate any solution μ of this problem by $\mu(s, Q_i, x_i, g_i)$.

Consider now the following routine.

Algorithm 5.4.1: Augmentation Algorithm

Input: $s_k \in \mathbb{E}^*$.

Iterate for $i = k-1, \ldots, 0$**:**

1. Compute $\mu_i := \mu(s_{i+1}, Q_i, x_i, g_i)$.
2. Set $s_i := s_{i+1} - \mu_i g_i$.

Lemma 5.4.1. *Let $\mu_0, \ldots, \mu_{k-1} \geq 0$ be generated by Algorithm 5.4.1. Then,*

$$\max_{x \in Q_0} \Big[\langle s_k, x \rangle + \sum_{i=0}^{k-1} \mu_i \langle g_i, x_i - x \rangle \Big] \leq \max_{x \in Q_k} \langle s_k, x \rangle.$$

Proof. Indeed, at every iteration $i = k-1, \ldots, 0$, we have

$$\begin{aligned} \max_{x \in Q_{i+1}} \langle s_{i+1}, x \rangle &\geq \max_{x \in \hat{Q}_i} \langle s_{i+1}, x \rangle = \max_{x \in Q_i} [\langle s_{i+1}, x \rangle + \mu_i \langle g_i, x_i - x \rangle] \\ &= \max_{x \in Q_i} \langle s_i, x \rangle + \mu_i \langle g_i, x_i \rangle. \end{aligned}$$

Summing up these inequalities for $i = 0, \dots, k-1$, we obtain

$$\begin{aligned}\max_{x \in Q_k} \langle s_k, x \rangle &\geq \max_{x \in Q_0} \langle s_0, x \rangle + \sum_{i=0}^{k-1} \mu_i \langle g_i, x_i \rangle \\ &= \max_{x \in Q_0} \Big[\langle s_k, x \rangle + \sum_{i=0}^{k-1} \langle g_i, x_i - x \rangle \Big],\end{aligned}$$

where the identity follows from the fact that $s_0 = s_k - \sum_{i=0}^{k-1} \mu_i g_i$. □

5.4.2 Methods with Preliminary Certificate

Let us apply the Augmentation Algorithm for building an accuracy semicertificate for Algorithm 5.2.1. We only consider those instances for which $\Gamma_k := \sum_{i=0}^{k-1} a_i \|g_i\|_* > 0$ so that the sliding gap Δ_k is well-defined:

$$\begin{aligned}\Delta_k &:= \max_{x \in \Omega_k} \frac{1}{\Gamma_k} [-\ell_k(x)] = \max_{x \in \Omega_k \cap L_k^-} \frac{1}{\Gamma_k} [-\ell_k(x)] \\ &= \max_{x \in \Omega_k \cap L_k^-} \frac{1}{\Gamma_k} \sum_{i=0}^{k-1} a_i \langle g_i, x_i - x \rangle.\end{aligned}$$

Recall that the vector $a := (a_0, \dots, a_{k-1})$ is called a preliminary semicertificate.

For technical reasons, it will be convenient to add the following termination criterion into Algorithm 5.2.1:

$$\text{Terminate Algorithm 5.2.1 at Step 2 if } U_k \leq \delta \|g_k\|_*, \tag{5.4.3}$$

where $\delta > 0$ is a fixed constant. Depending on whether this termination criterion has been satisfied at iteration k, we call it a *terminal* or *nonterminal* iteration, respectively.

Let $k \geq 1$ be an iteration of Algorithm 5.2.1. According to Lemma 5.2.1, the sets $Q_i := \Omega_i \cap L_i^-$ satisfy (5.4.1). Since the method has not been terminated during the course of the previous iterations, we have[4] $U_i > 0$ for all $0 \leq i \leq k-1$. Therefore, for any $0 \leq i \leq k-1$, there exists $x \in Q_i$ such that $\langle g_i, x - x_i \rangle < 0$. This guarantees the existence of dual multiplier in (5.4.2).

Let us apply Algorithm 5.4.1 to $s_k := -\sum_{i=0}^{k-1} a_i g_i$ in order to obtain

[4] Recall that $g_i \neq 0$ for all $i \geq 0$ by Remark 5.2.3.

dual multipliers $\mu := (\mu_0, \dots, \mu_{k-1})$. From Lemma 5.4.1, it follows that

$$\max_{x \in B(x_0,R)} \sum_{i=0}^{k-1} (a_i + \mu_i)\langle g_i, x_i - x\rangle \le \max_{x \in Q_k} \sum_{i=0}^{k-1} a_i \langle g_i, x_i - x\rangle = \Gamma_k \Delta_k$$

(note that $Q_0 = \Omega_0 \cap L_0^- = B(x_0, R)$). Thus, defining $\lambda := (\lambda_0, \dots, \lambda_{k-1})$ with $\lambda_i := a_i + \mu_i$ for all $0 \le i \le k-1$, we obtain

$$\Gamma_k(\lambda) \equiv \sum_{i=0}^{k-1} \lambda_i \|g_i\|_* \ge \sum_{i=0}^{k-1} a_i \|g_i\|_* \equiv \Gamma_k > 0$$

and

$$\delta_k(\lambda) \equiv \max_{x \in B(x_0,R)} \frac{1}{\Gamma_k(\lambda)} \sum_{i=0}^{k-1} \lambda_i \langle g_i, x_i - x\rangle \le \frac{\Gamma_k}{\Gamma_k(\lambda)} \Delta_k \le \Delta_k.$$

Thus, λ is a semicertificate whose gap on $B(x_0, R)$ is bounded from above by the sliding gap Δ_k.

If $k \ge 0$ is a terminal iteration, then, by the termination criterion and the definition of U_k (see Algorithm 5.2.1), we have

$$\max_{x \in \Omega_k \cap L_k^-} \frac{1}{\|g_k\|_*} \langle g_k, x_k - x\rangle \le \delta.$$

In this case, we apply Algorithm 5.4.1 to $s_k := -g_k$ to obtain dual multipliers $\mu_0, \dots, \mu_{k-1}$. By the same reasoning as above but with the vector $(0, \dots, 0, 1)$ instead of $(a_0, \dots, a_{k-1})$, we can obtain that $\delta_{k+1}(\lambda) \le \delta$, where $\lambda := (\mu_0, \dots, \mu_{k-1}, 1)$.

5.4.3 Standard Ellipsoid Method

In the standard Ellipsoid Method, there is no preliminary semicertificate. Therefore, we cannot apply the above procedure. However, in this method, it is still possible to generate an accuracy semicertificate although the corresponding procedure is slightly more involved. Let us now briefly describe this procedure and discuss how it differs from the previous approach. For details, we refer the reader to [125].

Let $k \ge 1$ be an iteration of the method. There are two main steps. The first step is to find a direction s_k, in which the "width" of the ellipsoid Ω_k

(see (5.3.3)) is minimal:

$$s_k \coloneqq \operatorname*{argmin}_{\|s\|_*=1} \max_{x,y\in\Omega_k} \langle s, x-y\rangle = \operatorname*{argmin}_{\|s\|_*=1} \Big[\max_{x\in\Omega_k} \langle s, x\rangle - \min_{x\in\Omega_k} \langle s, x\rangle \Big].$$

It is not difficult to see that s_k is given by any unit eigenvector[5] of the operator G_k, corresponding to the largest eigenvalue. For the corresponding minimal "width" of the ellipsoid, we have the following bound via the average radius (defined in (5.3.4)):

$$\max_{x,y\in\Omega_k} \langle s_k, x-y\rangle \le \rho_k, \tag{5.4.4}$$

where $\rho_k \coloneqq 2\operatorname{avrad}\Omega_k$. Recall that $\operatorname{avrad}\Omega_k \le \exp\big(-k/(2n^2)\big)R$ in view of (5.3.12).

At the second step, we apply Algorithm 5.4.1 two times with the sets $Q_i \coloneqq \Omega_i$: first, to the vector s_k to obtain dual multipliers $\mu \coloneqq (\mu_0, \dots, \mu_{k-1})$ and then to the vector $-s_k$ to obtain dual multipliers $\mu' \coloneqq (\mu'_0, \dots, \mu'_{k-1})$. By Lemma 5.4.1 and (5.4.4), we have

$$\max_{x\in B(x_0,R)} \Big[\langle s_k, x-x_k\rangle + \sum_{i=0}^{k-1} \mu_i \langle g_i, x_i - x\rangle \Big] \le \max_{x\in\Omega_k} \langle s_k, x-x_k\rangle \le \rho_k,$$

and

$$\max_{x\in B(x_0,R)} \Big[\langle s_k, x_k-x\rangle + \sum_{i=0}^{k-1} \mu'_i \langle g_i, x_i - x\rangle \Big] \le \max_{x\in\Omega_k} \langle s_k, x_k-x\rangle \le \rho_k$$

(note that $Q_0 = \Omega_0 = B(x_0, R)$). Consequently, for $\lambda \coloneqq \mu + \mu'$, we obtain

$$\max_{x\in B(x_0,R)} \sum_{i=0}^{k-1} \lambda_i \langle g_i, x_i - x\rangle \le 2\rho_k.$$

Finally, one can show that

$$\Gamma_k(\lambda) \equiv \sum_{i=0}^{k-1} \lambda_i \|g_i\|_* \ge \frac{r-\rho_k}{D},$$

where D is the diameter of Q and r is the maximal of the radii of Euclidean balls contained in Q. Thus, whenever $\rho_k < r$, λ is a semicertificate with the

[5] Here eigenvectors and eigenvalues are defined w.r.t. the operator B inducing the norm $\|\cdot\|$.

following gap on $B(x_0, R)$:

$$\delta_k(\lambda) \equiv \max_{x \in B(x_0,R)} \frac{1}{\Gamma_k(\lambda)} \sum_{i=0}^{k-1} \lambda_i \langle g_i, x_i - x \rangle \leq \frac{2\rho_k D}{r - \rho_k}.$$

Compared to the standard Ellipsoid Method, we see that, in the Subgradient Ellipsoid methods, the presence of the preliminary semicertificate removes the necessity in finding the minimal-"width" direction and requires only one run of the Augmentation Algorithm.

5.5 Implementation Details

5.5.1 Explicit Representations

In the implementation of Algorithm 5.2.1, instead of the operators G_k, it is better to work with their inverses $H_k \coloneqq G_k^{-1}$. Applying the Sherman-Morrison formula to (5.2.3), we obtain the following update rule for H_k:

$$H_{k+1} = H_k - \frac{b_k H_k g_k g_k^* H_k}{1 + b_k \langle g_k, H_k g_k \rangle}, \quad k \geq 0. \tag{5.5.1}$$

Let us now obtain an explicit formula for the next test point x_{k+1}. This has already been partly done in the proof of Lemma 5.2.2. Indeed, recall that x_{k+1} is the minimizer of the function $\psi_{k+1}(x)$. From (5.2.6), we see that $x_{k+1} = x_k - (a_k + \frac{1}{2} b_k U_k) H_{k+1} g_k$. Combining it with (5.5.1), we obtain

$$x_{k+1} = x_k - \frac{a_k + \frac{1}{2} b_k U_k}{1 + b_k \langle g_k, H_k g_k \rangle} H_k g_k, \quad k \geq 0. \tag{5.5.2}$$

Finally, one can obtain the following explicit representations for L_k^- and Ω_k:

$$\begin{aligned} L_k^- &= \{x \in \mathbb{E} : \langle c_k, x \rangle \leq \sigma_k\}, \\ \Omega_k &= \{x \in \mathbb{E} : \|x - z_k\|^2_{H_k^{-1}} \leq D_k\}, \end{aligned} \tag{5.5.3}$$

where

$$\begin{aligned} c_0 \coloneqq 0, \quad \sigma_0 \coloneqq 0, \quad c_{k+1} \coloneqq c_k + a_k g_k, \quad \sigma_{k+1} \coloneqq \sigma_k + a_k \langle g_k, x_k \rangle, \\ z_k \coloneqq x_k - H_k c_k, \quad D_k \coloneqq R_k^2 + 2(\sigma_k - \langle c_k, x_k \rangle) + \langle c_k, H_k c_k \rangle \end{aligned} \tag{5.5.4}$$

for any $k \geq 0$. Indeed, recalling the definition of functions ℓ_k, we see that

$\ell_k(x) = \langle c_k, x\rangle - \sigma_k$ for all $x \in \mathbb{E}$. Therefore,

$$L_k^- \equiv \{x : \ell_k(x) \le 0\} = \{x : \langle c_k, x\rangle \le \sigma_k\}.$$

Further, by Lemma 5.2.2,

$$\Omega_k = \{x : \langle c_k, x\rangle + \tfrac{1}{2}\|x - x_k\|_{G_k}^2 \le \tfrac{1}{2}R_k^2 + \sigma_k\}.$$

Note that

$$\langle c_k, x\rangle + \tfrac{1}{2}\|x - x_k\|_{G_k}^2 = \tfrac{1}{2}\|x - z_k\|_{G_k}^2 + \langle c_k, x_k\rangle - \tfrac{1}{2}(\|c_k\|_{G_k}^*)^2$$

for any $x \in \mathbb{E}$. Hence, $\Omega_k = \{x : \tfrac{1}{2}\|x - z_k\|_{G_k}^2 \le \tfrac{1}{2}D_k\}$.

Remark 5.5.1. Now we can justify the claim made in Section 5.3.2 that Algorithm 5.2.1 with parameters (5.2.11), (5.3.2) and (5.3.11) is the standard Ellipsoid Method. Indeed, from (5.2.11) and (5.3.3), we see that

$$b_k = \frac{\gamma}{\langle g_k, H_k g_k\rangle}, \qquad U_k = R_k\langle g_k, H_k g_k\rangle^{1/2}.$$

Also, in view of (5.3.11),

$$\frac{\gamma}{1+\gamma} = \frac{2}{n+1}.$$

Hence, by (5.5.2) and (5.5.1),

$$\begin{aligned} x_{k+1} &= x_k - \frac{R_k}{n+1}\frac{H_k g_k}{\langle g_k, H_k g_k\rangle^{1/2}}, \\ H_{k+1} &= H_k - \frac{2}{n+1}\frac{H_k g_k g_k^* H_k}{\langle g_k, H_k g_k\rangle}, \qquad k \ge 0. \end{aligned} \tag{5.5.5}$$

Further, according to (5.3.7) and (5.3.11), for any $k \ge 0$, we have

$$R_k^2 = q^k R^2,$$

where

$$q = 1 + \frac{1}{(n-1)(n+1)} = \frac{n^2}{n^2-1}.$$

Thus, method (5.5.5) indeed coincides with the standard Ellipsoid Method (5.0.2) under the change of variables $W_k := R_k^2 H_k$.

5.5.2 Computing Support Function

To calculate U_k in Algorithm 5.2.1, we need to compute the following quantity (see (5.5.3)):

$$U_k = \max_x \{\langle g_k, x_k - x\rangle : \|x - z_k\|^2_{H_k^{-1}} \le D_k, \langle c_k, x\rangle \le \sigma_k\}.$$

Let us discuss how to do this.

First, let us introduce the following support function to simplify our notation:

$$\xi(H, s, a, \beta) \coloneqq \max_x \{\langle s, x\rangle : \|x\|^2_{H^{-1}} \le 1, \langle a, x\rangle \le \beta\},$$

where $H \in \mathcal{S}_{++}(\mathbb{E}^*, \mathbb{E})$, $s, a \in \mathbb{E}^*$ and $\beta \in \mathbb{R}$. In this notation, assuming that $D_k > 0$, we have

$$U_k = \langle g_k, x_k - z_k\rangle + \xi(D_k H_k, -g_k, c_k, \sigma_k - \langle c_k, z_k\rangle).$$

Let us show how to compute $\xi(H, s, a, \beta)$. Dualizing the linear constraint, we obtain

$$\xi(H, s, a, \beta) = \min_{\tau \ge 0} \big[\|s - \tau a\|^*_{H^{-1}} + \tau\beta\big], \tag{5.5.6}$$

provided that there exists some $x \in \mathbb{E}$ such that $\|x\|_{H^{-1}} < 1$, $\langle a, x\rangle \le \beta$ (Slater condition). One can show that problem (5.5.6) has the following solution (see Lemma 5.B.2):

$$\tau(H, s, a, \beta) \coloneqq \begin{cases} 0, & \text{if } \langle a, Hs\rangle \le \beta\|s\|^*_{H^{-1}}, \\ u(H, s, a, \beta), & \text{otherwise,} \end{cases} \tag{5.5.7}$$

where $u(H, s, a, \beta)$ is the unconstrained minimizer of the objective function in (5.5.6).

Let us present an explicit formula for $u(H, s, a, \beta)$. For future use, it will be convenient to write down this formula in a slightly more general form for the following multidimensional variant of problem (5.5.6):

$$\min_{u \in \mathbb{R}^m} \big[\|s - Au\|^*_{H^{-1}} + \langle u, b\rangle\big], \tag{5.5.8}$$

where $s \in \mathbb{E}^*$, $H \in \mathcal{S}_{++}(\mathbb{E}^*, \mathbb{E})$, $A\colon \mathbb{R}^m \to \mathbb{E}^*$ is a linear operator with trivial kernel and $b \in \mathbb{R}^m$, $\langle b, (A^* H A)^{-1} b\rangle < 1$. It is not difficult to show

that problem (5.5.8) has the following unique solution (see Lemma 5.B.1):

$$\begin{aligned} u(H, s, A, b) &:= (A^* H A)^{-1}(A^* s - rb), \\ r &:= \sqrt{\frac{\langle s, Hs\rangle - \langle s, A(A^* H A)^{-1} A^* s\rangle}{1 - \langle b, (A^* H A)^{-1} b\rangle}}. \end{aligned} \tag{5.5.9}$$

Note that, in order for the above approach to work, we need to guarantee that the sets Ω_k and L_k^- satisfy a certain regularity condition, namely, $\operatorname{int}\Omega_k \cap L_k^- \neq \varnothing$. This condition can be easily fulfilled by adding into Algorithm 5.2.1 the termination criterion (5.4.3).

Lemma 5.5.2. *Consider Algorithm 5.2.1 with termination criterion (5.4.3). Then, at each iteration $k \geq 0$, at the beginning of Step 2, we have $\operatorname{int}\Omega_k \cap L_k^- \neq \varnothing$. Moreover, if k is a nonterminal iteration, we also have $\langle g_k, x - x_k\rangle \leq 0$ for some $x \in \operatorname{int}\Omega_k \cap L_k^-$.*

Proof. Note that $\operatorname{int}\Omega_0 \cap L_0^- = \operatorname{int} B(x_0, R) \neq \varnothing$. Now suppose $\operatorname{int}\Omega_k \cap L_k^- \neq \varnothing$ for some nonterminal iteration $k \geq 0$. Denote $P_k^- := \{x \in \mathbb{E} : \langle g_k, x - x_k\rangle \leq 0\}$. Since iteration k is nonterminal, $U_k > 0$ and hence $\Omega_k \cap L_k^- \cap \operatorname{int} P_k^- \neq \varnothing$. Combining it with the fact that $\operatorname{int}\Omega_k \cap L_k^- \neq \varnothing$, we obtain $\operatorname{int}\Omega_k \cap L_k^- \cap \operatorname{int} P_k^- \neq \varnothing$ and, in particular, $\operatorname{int}\Omega_k \cap L_k^- \cap P_k^- \neq \varnothing$. At the same time, slightly modifying the proof of Lemma 5.2.1 (using that $\operatorname{int}\Omega_i = \{x \in \mathbb{E} : \omega_i(x) < \frac{1}{2}R^2\}$ for any $i \geq 0$ since ω_i is a strictly convex quadratic function), it is not difficult to show that $\operatorname{int}\Omega_k \cap L_k^- \cap P_k^- \subseteq \operatorname{int}\Omega_{k+1} \cap L_{k+1}^-$. Thus, $\operatorname{int}\Omega_{k+1} \cap L_{k+1}^- \neq \varnothing$, and we can continue by induction. □

5.5.3 Computing Dual Multipliers

Recall from Section 5.4 that the procedure for generating an accuracy semicertificate for Algorithm 5.2.1 requires one to repeatedly carry out the following operation: given $s \in \mathbb{E}^*$ and some iteration number $i \geq 0$, compute a dual multiplier $\mu \geq 0$ such that

$$\max_{x \in \Omega_i \cap L_i^-} \{\langle s, x\rangle : \langle g_i, x - x_i\rangle \leq 0\} = \max_{x \in \Omega_i \cap L_i^-} \big[\langle s, x\rangle + \mu\langle g_i, x_i - x\rangle\big].$$

This can be done as follows.

First, using (5.5.3), let us rewrite the above primal problem more ex-

plicitly:

$$\max_x \{\langle s, x\rangle : \|x - z_i\|^2_{H_i^{-1}} \le D_i, \langle c_i, x\rangle \le \sigma_i, \langle g_i, x - x_i\rangle \le 0\}.$$

Our goal is to dualize the second linear constraint and find the corresponding multiplier. However, for the sake of symmetry, it is better to dualize both linear constraints, find the corresponding multipliers and then keep only the second one.

Let us simplify our notation by introducing the following problem:

$$\max_x \{\langle s, x\rangle : \|x\|_{H^{-1}} \le 1, \langle a_1, x\rangle \le b_1, \langle a_2, x\rangle \le b_2\}, \tag{5.5.10}$$

where $H \in \mathcal{S}_{++}(\mathbb{E}^*, \mathbb{E})$, $s, a_1, a_2 \in \mathbb{E}^*$ and $b_1, b_2 \in \mathbb{R}$. Clearly, our original problem can be transformed into this form by setting $H := D_i H_i$, $a_1 := c_i$, $a_2 := g_i$, $b_1 := \sigma_i - \langle c_i, z_i\rangle$, $b_2 := \langle g_i, x_i - z_i\rangle$. Note that this transformation does not change the dual multipliers.

Dualizing the linear constraints in (5.5.10), we obtain the following dual problem:

$$\min_{\mu \in \mathbb{R}^2_+} \left[\|s - \mu_1 a_1 - \mu_2 a_2\|^*_{H^{-1}} + \mu_1 b_1 + \mu_2 b_2\right], \tag{5.5.11}$$

which is solvable provided the following Slater condition holds:

$$\exists x \in \mathbb{E}\colon \|x\|_{H^{-1}} < 1, \langle a_1, x\rangle \le b_1, \langle a_2, x\rangle \le b_2. \tag{5.5.12}$$

Note that condition (5.5.12) can be ensured by adding termination criterion (5.4.3) into Algorithm 5.2.1 (see Lemma 5.5.2).

A solution of problem (5.5.11) can be found using Algorithm 5.5.1. In this routine, $\tau(\cdot)$, $\xi(\cdot)$ and $u(\cdot)$ are the auxiliary operations, defined in Section 5.5.2, and $A := (a_1, a_2)$ is the linear operator $Au := u_1 a_1 + u_2 a_2$ acting from $\mathbb{R}^2$ to $\mathbb{E}^*$. The correctness of Algorithm 5.5.1 is proved in Theorem 5.B.4.

Algorithm 5.5.1: Computing Dual Multipliers

1. Compute $\tau_1 := \tau(H, s, a_1, b_1)$ and $\tau_2 := \tau(H, s, a_2, b_2)$.
Compute $\xi_1 := \xi(H, a_2, a_1, b_1)$ and $\xi_2 := \xi(H, a_1, a_2, b_2)$.

2. If $\xi_1 \le b_2$, return $(\tau_1, 0)$. Else if $\xi_2 \le b_1$, return $(0, \tau_2)$.

3. Else if $\langle a_2, H(s - \tau_1 a_1)\rangle \le b_2 \|s - \tau_1 a_1\|^*_{H^{-1}}$, return $(\tau_1, 0)$.
Else if $\langle a_1, H(s - \tau_2 a_2)\rangle \le b_1 \|s - \tau_2 a_2\|^*_{H^{-1}}$, return $(0, \tau_2)$.

4. Else return $u := u(H, s, A, b)$, where $A := (a_1, a_2)$, $b := (b_1, b_2)^T$.

5.5.4 Time and Memory Requirements

Let us discuss the time and memory requirements of Algorithm 5.2.1, taking into account the previously mentioned implementation details.

The main objects in Algorithm 5.2.1, which need to be stored and updated between iterations, are the test points x_k, matrices H_k, scalars R_k, vectors c_k and scalars σ_k, see (5.5.2), (5.5.1), (5.2.4) and (5.5.4) for the corresponding updating formulas. To store all these objects, we need $O(n^2)$ memory.

Consider now what happens at each iteration k. First, we compute U_k. For this, we calculate z_k and D_k according to (5.5.4) and then perform the calculations described in Section 5.5.2. The most difficult operation there is computing the matrix-vector product, which takes $O(n^2)$ time. After that, we calculate the coefficients a_k and b_k according to (5.2.11), where α_k, θ and γ are certain scalars, easily computable for all main instances of Algorithm 5.2.1 (see Sections 5.3.1–5.3.4). The most expensive step there is computing the norm $\|g_k\|^*_{G_k}$, which can be done in $O(n^2)$ operations by evaluating the product $H_k g_k$. Finally, we update our main objects, which takes $O(n^2)$ time.

Thus, each iteration of Algorithm 5.2.1 has $O(n^2)$ time and memory complexities, exactly as in the standard Ellipsoid Method.

Now let us analyze the complexity of the auxiliary procedure from Section 5.4 for converting a preliminary semicertificate into a semicertificate. The main operation in this procedure is running Algorithm 5.4.1, which iterates "backwards", computing some dual multiplier μ_i at each iteration $i = k-1, \dots, 0$. Using the approach from Section 5.5.3, we can compute μ_i in $O(n^2)$ time, provided that the objects x_i, g_i, H_i, z_i, D_i, c_i, σ_i are stored in memory. Note, however, that, in contrast to the "forward" pass, when iterating "backwards", there is no way to efficiently recompute all these

objects without storing in memory a certain "history" of the main process from iteration 0 up to k. The simplest choice is to keep in this "history" all the objects mentioned above, which requires $O(kn^2)$ memory. A slightly more efficient idea is to keep the matrix-vector products $H_i g_i$ instead of H_i and then use (5.5.1) to recompute H_i from H_{i+1} in $O(n^2)$ operations. This allows us to reduce the size of the "history" down to $O(kn)$ while still keeping the $O(kn^2)$ total time complexity of the auxiliary procedure. Note that these estimates are exactly the same as those for the best currently known technique for generating accuracy certificates in the standard Ellipsoid Method [125]. In particular, if we generate a semicertificate only once at the very end, then the time complexity of our procedure is comparable to that of running the standard Ellipsoid Method without computing any certificates. Alternatively, as suggested in [125], one can generate semicertificates, say, every $2, 4, 8, 16, \ldots$ iterations. Then, the total "overhead" of the auxiliary procedure for generating semicertificates will be comparable to the time complexity of the method itself.

5.6 Discussion

In this chapter, we have presented a new algorithm—the Subgradient Ellipsoid Method—for solving general nonsmooth problems with convex structure. This algorithm can be seen as the combination of the "dimension-dependent" Ellipsoid Method, which is efficient for small-dimensional problems, and the "dimension-independent" Subgradient Method, which is much more efficient for large-scale problems.

Compared to the Ellipsoid Method, the Subgradient Ellipsoid Method has virtually the same complexity of each iteration. However, it is more robust with respect to the space dimension n. Furthermore, the procedure for generating accuracy certificates in the Ellipsoid Subgradient Method is slightly simpler.

Our developments can be considered as a first step towards constructing universal methods for nonsmooth problems with convex structure. Such methods could significantly improve the practical efficiency of solving various applied problems.

Let us discuss some open questions. First, the convergence rate estimate of the Subgradient Ellipsoid Method with time-varying coefficients contains an extra factor proportional to the logarithm of the iteration counter. We have seen that this logarithmic factor has its roots in the Subgradi-

ent Method. However, as discussed in Remark 5.3.1, for the Subgradient Method, this issue can be easily resolved by allowing projections onto the feasible set and working with "truncated" gaps. An even better alternative, which does not require any of this machinery, is to use Dual Averaging [130] instead of the Subgradient Method. It is an interesting question whether one can combine the Dual Averaging with the Ellipsoid Method similarly to how we have combined the Subgradient and Ellipsoid methods.

Second, the convergence rate estimate, which we have obtained for the Subgradient Ellipsoid Method, is not continuous in the space dimension n. Indeed, for small values of the iteration counter k, this estimate behaves as that of the Subgradient Method and then, at some moment (around n^2), it switches to the estimate of the Ellipsoid Method. As discussed at the end of Section 5.3.4, there exists some "small" gap between these two estimates around the switching moment. Nevertheless, the method itself is continuous in n and does not contain any explicit switching rules. Therefore, there should be some continuous convergence rate estimate for the method, and it is an open question to find it.

Finally, besides the Ellipsoid Method, there exist other "dimension-dependent" methods, e.g., the Center-of-Gravity Method[6] [107, 142], the Inscribed Ellipsoid Method [179], the Circumscribed Simplex Method [23], etc. Similarly, the Subgradient Method is not the only "dimension-independent" method and there exist numerous alternatives which are better suited for certain problem classes, e.g., the Fast Gradient Method [127] for Smooth Convex Optimization or various methods for Stochastic Programming [56, 57, 101, 124]. Of course, it is interesting to consider different combinations of the aforementioned "dimension-dependent" and "dimension-independent" methods. In this regard, it is also worth mentioning the works [21, 22], where the authors propose new variants of gradient-type methods for smooth strongly convex minimization problems inspired by the geometric construction of the Ellipsoid Method.

5.A Proof of Lemma 5.3.2

Proof. Everywhere in the proof, we assume that the parameter c is fixed and drop all the indices related to it.

Let us show that ζ_p is a convex function. Indeed, the function $\omega\colon \mathbb{R} \times$

[6] Although this method is not practical, it is still interesting from an academic point of view.

$\mathbb{R}_{++} \to \mathbb{R}$ defined by $\omega(x,t) \coloneqq x^2/t$ is convex. Hence, the function q, defined in (5.3.6), is also convex. Further, since ω is increasing in its first argument on $\mathbb{R}_+$, the function $\omega_p \colon \mathbb{R}_+ \times \mathbb{R}_{++} \to \mathbb{R}$ defined by $\omega_p(x,t) \coloneqq x^p/t$ is also convex as the composition of ω with the mapping $(x,t) \mapsto (x^{p/2},t)$, whose first component is convex (since $p \geq 2$) and the second one is affine. Note that ω_p is increasing in its first argument. Hence, ζ_p is indeed a convex function as the composition of ω_p with the mapping $\gamma \mapsto \big(q(\gamma), 1+\gamma\big)$, whose first part is convex and the second one is affine.

Differentiating, for any $\gamma > 0$, we obtain

$$\begin{aligned}\zeta_p'(\gamma) &= \frac{p[q(\gamma)]^{p-1}q'(\gamma)(1+\gamma) - [q(\gamma)]^p}{(1+\gamma)^2} \\ &= \frac{[q(\gamma)]^{p-1}\big(pq'(\gamma)(1+\gamma) - q(\gamma)\big)}{(1+\gamma)^2}.\end{aligned}$$

Hence, the minimizers of ζ_p are exactly solutions to the following equation:

$$pq'(\gamma)(1+\gamma) = q(\gamma). \tag{5.A.1}$$

Note, from (5.3.6), that

$$q'(\gamma) = \frac{c[2\gamma(1+\gamma) - \gamma^2]}{2(1+\gamma)^2} = \frac{c\gamma(2+\gamma)}{2(1+\gamma)^2}.$$

Hence, (5.A.1) can be written as

$$cp\gamma(2+\gamma) = 2(1+\gamma) + c\gamma^2$$

or, equivalently, as

$$c(p-1)\gamma^2 + 2(cp-1)\gamma = 2.$$

Clearly, $\gamma = 0$ is not a solution of this equation. Making the change of variables $\gamma = 2/u$, $u \neq 0$, we come the quadratic equation

$$u^2 - 2(cp-1)u = 2c(p-1)$$

or, equivalently, to

$$[u-(cp-1)]^2 = 2c(p-1) + (cp-1)^2 = c^2p^2 - (2c-1).$$

This equation has two solutions:

$$u_1 := cp - 1 + \sqrt{c^2p^2 - (2c-1)}, \qquad u_2 := cp - 1 - \sqrt{c^2p^2 - (2c-1)}.$$

Note that

$$u_2 \geq cp - 1 - \sqrt{c^2p^2 + 1} \geq cp - 1 - (cp+1) = -2.$$

Hence, $\gamma_2 := 2/u_2 \leq -1$ cannot be a minimizer of ζ_p. Consequently, only u_1 is an acceptable solution (note that $u_1 > 0$ in view of our assumptions on c and p). Thus, (5.3.10) is proved.

Let us show that $\gamma(p)$ belongs to the interval specified in (5.3.10). For this, we need to prove that $1 \leq cp\gamma(p) \leq 2$. Note that the function

$$h_a(t) := \frac{t}{\sqrt{t^2 - a} + t - 1},$$

where $a \geq 0$, is decreasing in t. Indeed, $[h_a(t)]^{-1} = \sqrt{1 - \frac{a}{t^2}} - \frac{1}{t} + 1$ is an increasing function in t. Hence,

$$cp\gamma(p) = 2h_{2c-1}(cp) \geq 2 \lim_{t\to\infty} h_{2c-1}(t) = 1.$$

On the other hand, using that $p \geq 2$ and denoting $\alpha := 2c \geq 1$, we get

$$cp\gamma(p) = 2h_{\alpha-1}(cp) \leq 2g(\alpha),$$

where

$$g(\alpha) := h_{\alpha-1}(\alpha) = \frac{\alpha}{\sqrt{\alpha^2 - \alpha + 1} + \alpha - 1}.$$

Note that g is decreasing in α. Indeed, denoting $\tau := 1/\alpha \in (0, 1]$, we get $[g(\alpha)]^{-1} = \sqrt{1 - \tau + \tau^2} - \tau + 1$, which is a decreasing function in τ. Thus,

$$cp\gamma(p) \leq 2g(1) = 2.$$

It remains to prove that

$$\zeta_p(\gamma(p)) \leq \exp\bigl(-1/(2cp)\bigr).$$

Let $\varphi\colon [2, +\infty) \to \mathbb{R}$ be the function

$$\varphi(p) := -\ln \zeta_p\bigl(\gamma(p)\bigr) = \ln\bigl(1 + \gamma(p)\bigr) - p \ln q\bigl(\gamma(p)\bigr). \tag{5.A.2}$$

We need to show that

$$\varphi(p) \geq \frac{1}{2cp}$$

for all $p \geq 2$ or, equivalently, that the function $\chi\colon (0, \frac{1}{2}] \to \mathbb{R}$ defined by

$$\chi(\tau) \coloneqq \varphi(\tau^{-1})$$

satisfies

$$\chi(\tau) \geq \frac{\tau}{2c}$$

for all $\tau \in (0, \frac{1}{2}]$. For this, it suffices to show that χ is convex,

$$\lim_{\tau\to 0} \chi(\tau) = 0, \qquad \lim_{\tau\to 0} \chi'(\tau) = \frac{1}{2c}.$$

Differentiating, we see that, for all $\tau \in (0, \frac{1}{2}]$,

$$\chi'(\tau) = -\tau^{-2}\varphi'(\tau^{-1}), \qquad \chi''(\tau) = 2\tau^{-3}\varphi'(\tau^{-1}) + \tau^{-4}\varphi''(\tau^{-1}).$$

Thus, we need to justify that

$$2\varphi'(p) + p\varphi''(p) \geq 0 \tag{5.A.3}$$

for all $p \geq 2$ and that

$$\lim_{p\to\infty} \varphi(p) = 0, \qquad \lim_{p\to\infty} [-p^2\varphi'(p)] = \frac{1}{2c}. \tag{5.A.4}$$

Let $p \geq 2$ be arbitrary. Differentiating and using (5.A.1), we obtain

$$\begin{aligned} \varphi'(p) &= \frac{\gamma'(p)}{1+\gamma(p)} - \ln q\big(\gamma(p)\big) - \frac{pq'\big(\gamma(p)\big)\gamma'(p)}{q\big(\gamma(p)\big)} = -\ln q\big(\gamma(p)\big), \\ \varphi''(p) &= -\frac{q'(\gamma(p))\gamma'(p)}{q(\gamma(p))} = -\frac{\gamma'(p)}{p\big(1+\gamma(p)\big)}. \end{aligned} \tag{5.A.5}$$

Therefore,

$$2\varphi'(p) + p\varphi''(p) = -2\ln q(\gamma(p)) - \frac{\gamma'(p)}{1+\gamma(p)} \geq -\frac{c\gamma^2(p) + \gamma'(p)}{1+\gamma(p)},$$

where the inequality follows from (5.3.6) and the fact that $\ln(1+\tau) \leq \tau$ for

any $\tau > -1$. Thus, to show (5.A.3), we need to prove that

$$-\gamma'(p) \geq c\gamma^2(p)$$

or, equivalently, that

$$\frac{d}{dp}\frac{1}{\gamma(p)} \geq c.$$

But this is immediate. Indeed, using (5.3.10), we obtain

$$\frac{d}{dp}\frac{1}{\gamma(p)} = \frac{c}{2}\Big(\frac{cp}{\sqrt{c^2p^2 - (2c-1)}} + 1\Big) \geq c$$

since the function $\tau \mapsto \tau/\sqrt{\tau^2 - 1}$ is decreasing. Thus, (5.A.3) is proved.

It remains to show (5.A.4). From (5.3.10), we see that $\gamma(p) \to 0$ and $p\gamma(p) \to c^{-1}$ as $p \to \infty$. Hence, using (5.3.6), we obtain

$$\lim_{p\to\infty} p^2 \ln q\big(\gamma(p)\big) = \lim_{p\to\infty} \frac{cp^2\gamma^2(p)}{2\big(1+\gamma(p)\big)} = \frac{c}{2}\lim_{p\to\infty} p^2\gamma^2(p) = \frac{1}{2c}.$$

Consequently, in view of (5.A.2) and (5.A.5), we have

$$\lim_{p\to\infty} \varphi(p) = \lim_{p\to\infty} \big[\ln\big(1+\gamma(p)\big) - p\ln q\big(\gamma(p)\big)\big] = 0,$$
$$\lim_{p\to\infty} [-p^2\varphi'(p)] = \lim_{p\to\infty} p^2 \ln q\big(\gamma(p)\big) = \frac{1}{2c},$$

which is exactly (5.A.4). □

5.B Support Function and Dual Multipliers: Proofs

For brevity, everywhere in this section, we write $\|\cdot\|$ and $\|\cdot\|_*$ instead of $\|\cdot\|_{H^{-1}}$ and $\|\cdot\|^*_{H^{-1}}$, respectively. We also denote $B_0 := \{x \in \mathbb{E} : \|x\| \leq 1\}$.

5.B.1 Auxiliary Operations

Lemma 5.B.1. *Let $s \in \mathbb{E}^*$, let $A\colon \mathbb{R}^m \to \mathbb{E}^*$ be a linear operator with trivial kernel and let $b \in \mathbb{R}^m$, $\langle b, (A^*HA)^{-1}b\rangle < 1$. Then, problem (5.5.8) has a unique solution given by (5.5.9).*

Proof. Note that the sublevel sets of the objective function in (5.5.8) are bounded:

$$\begin{aligned}\|s - Au\|_* + \langle u, b\rangle &\geq \|Au\|_* - \|s\|_* + \langle u, b\rangle \\ &\geq (1 - \langle b, (A^*HA)^{-1}b\rangle^{1/2})\|Au\|_* - \|s\|_*\end{aligned}$$

for all $u \in \mathbb{R}^m$. Hence, problem (5.5.8) has a solution.

Let $u \in \mathbb{R}^m$ be a solution of problem (5.5.8). If $s = Au$, then $u = (A^*HA)^{-1}A^*s$, which coincides with the solution given by (5.5.9) (note that, in this case, $r = 0$).

Now suppose $s \neq Au$. Then, from the first-order optimality condition, we obtain that $b = A^*(s - Au)/\rho$, where $\rho := \|s - Au\|_* > 0$. Hence, $u = (A^*HA)^{-1}(A^*s - \rho b)$ and

$$\begin{aligned}\rho^2 = \|s - Au\|_*^2 &= \|s\|_*^2 - 2\langle A^*s, u\rangle + \langle A^*HAu, u\rangle \\ &= \|s\|_*^2 - 2\langle A^*s, (A^*HA)^{-1}(A^*s - \rho b)\rangle \\ &\quad + \langle A^*s - \rho b, (A^*HA)^{-1}(A^*s - \rho b)\rangle \\ &= \|s\|_*^2 - \langle s, A(A^*HA)^{-1}A^*s\rangle + \rho^2\langle b, (A^*HA)^{-1}b\rangle.\end{aligned}$$

Thus, $\rho = r$ and $u = u(H, s, A, b)$ given by (5.5.9). □

Lemma 5.B.2. *Let $s, a \in \mathbb{E}^*$, $\beta \in \mathbb{R}$ be such that $\langle a, x\rangle \leq \beta$ for some $x \in \operatorname{int} B_0$. Then, problem (5.5.6) has a solution given by (5.5.7). Moreover, this solution is unique if $\beta < \|a\|_*$.*

Proof. Let $\varphi\colon \mathbb{R} \to \mathbb{R}$ be the function $\varphi(\tau) := \|s - \tau a\|_* + \tau\beta$. By our assumptions, $\beta > -\|a\|_*$ if $a \neq 0$ and $\beta \geq 0$ if $a = 0$. If additionally $\beta < \|a\|_*$, then $|\beta| < \|a\|_*$.

If $s = 0$, then $\varphi(\tau) = \tau(\|a\|_* + \beta) \geq \varphi(0)$ for all $\tau \geq 0$, so 0 is a solution of (5.5.6). Clearly, this solution is unique when $\beta < \|a\|_*$ because then $|\beta| < \|a\|_*$.

From now on, suppose $s \neq 0$. Then, φ is differentiable at 0 with $\varphi'(0) = \beta - \langle a, s\rangle/\|s\|_*$. If $\langle a, s\rangle \leq \beta\|s\|_*$, then $\varphi'(0) \geq 0$, so 0 is a solution of (5.5.6). Note that this solution is unique if $\langle a, s\rangle < \beta\|s\|_*$ because then $\varphi'(0) > 0$, i.e., φ is strictly increasing on $\mathbb{R}_+$.

Suppose $\langle a, s\rangle > \beta\|s\|_*$. Then, $\beta < \|a\|_*$ and thus $|\beta| < \|a\|_*$. Note that, for any $\tau \geq 0$, we have $\varphi(\tau) \geq \tau(\|a\|_* + \beta) - \|s\|_*$. Hence, the sublevel sets of φ, intersected with $\mathbb{R}_+$, are bounded, so problem (5.5.6) has a solution. Since $\varphi'(0) < 0$, any solution of (5.5.6) is strictly positive and so must be

a solution of problem (5.5.8) for $A \coloneqq a$ and $b \coloneqq \beta$. But, by Lemma 5.B.1, the latter solution is unique and equals $u(H, s, a, \beta)$.

We have proved that (5.5.7) is indeed a solution of (5.5.6). Moreover, when $\langle a, s\rangle \neq \beta\|s\|_*$, we have shown that this solution is unique. It remains to prove the uniqueness of solution when $\langle a, s\rangle = \beta\|s\|_*$, assuming additionally that $\beta < \|a\|_*$. But this is simple. Indeed, by our assumptions, $|\beta| < \|a\|_*$, so $|\langle a, s\rangle| = |\beta|\|s\|_* < \|a\|_*\|s\|_*$. Hence, a and s are linearly independent. But then φ is strictly convex and thus its minimizer is unique. □

5.B.2 Computation of Dual Multipliers

In this section, we prove the correctness of Algorithm 5.5.1.

For $s \in \mathbb{E}^*$, let $X(s)$ be the subdifferential of $\|\cdot\|_*$ at the point s:

$$X(s) \coloneqq \begin{cases} \{Hs/\|s\|_*\}, & \text{if } s \neq 0, \\ B_0, & \text{if } s = 0. \end{cases} \tag{5.B.1}$$

Clearly, $X(s) \subseteq B_0$ for any $s \in \mathbb{E}^*$. When $s \neq 0$, we denote the unique element of $X(s)$ by $x(s)$.

Let us formulate a convenient optimality condition.

Lemma 5.B.3. *Let A be the linear operator from $\mathbb{R}^m$ to $\mathbb{E}^*$, defined by $Au \coloneqq \sum_{i=1}^m u_i a_i$, where $a_1, \dots, a_m \in \mathbb{E}^*$, and let $b \in \mathbb{R}^m$, $s \in \mathbb{E}^*$. Then, $\mu^* \in \mathbb{R}^m_+$ is a minimizer of the function*

$$\psi(\mu) \coloneqq \|s - A\mu\|_* + \langle \mu, b\rangle$$

over $\mathbb{R}^m_+$ if and only if

$$X(s - A\mu^*) \cap L_1(\mu_1^*) \dots L_m(\mu_m^*) \neq \varnothing,$$

where, for each $1 \leq i \leq m$ and $\tau \geq 0$, we denote

$$L_i(\tau) \coloneqq \begin{cases} \{x \in \mathbb{E} : \langle a_i, x\rangle \leq b_i\}, & \text{if } \tau = 0, \\ \{x \in \mathbb{E} : \langle a_i, x\rangle = b_i\}, & \text{if } \tau > 0. \end{cases}$$

Proof. Indeed, the standard optimality condition for a convex function over the nonnegative orthant is as follows: $\mu^* \in \mathbb{R}^m_+$ is a minimizer of ψ on $\mathbb{R}^m_+$ if and only if there exists $g^* \in \partial\psi(\mu^*)$ such that $g_i^* \geq 0$ and $g_i^*\mu_i^* = 0$ for

all $1 \le i \le m$. It remains to note that $\partial\psi(\mu^*) = b - A^*X(s - A\mu^*)$. □

Theorem 5.B.4. *Algorithm 5.5.1 is well-defined and returns a solution of (5.5.11).*

Proof. i. For each $i = 1, 2$ and $\tau \ge 0$, denote $L_i^- := \{x \in \mathbb{E} : \langle a_i, x\rangle \le b_i\}$, $L_i := \{x \in \mathbb{E} : \langle a_i, x\rangle = b_i\}$, $L_i(\tau) := L_i^-$, if $\tau = 0$, and $L_i(\tau) := L_i$, if $\tau > 0$.

ii. From (5.5.12) and Lemma 5.B.2, it follows that Step 1 is well-defined and, for each $i = 1, 2$, τ_i is a solution of (5.5.6) with parameters (s, a_i, b_i). Hence, by Lemma 5.B.3,

$$X(s - \tau_i a_i) \cap L_i(\tau_i) \ne \varnothing, \quad i = 1, 2. \tag{5.B.2}$$

iii. Consider Step 2. Note that the condition $\xi_1 \le b_2$ is equivalent to $B_0 \cap L_1^- \subseteq L_2^-$ since $\xi_1 = \max_{x \in B_0 \cap L_1^-} \langle a_2, x\rangle$. If $B_0 \cap L_1^- \subseteq L_2^-$, then, by (5.B.2), $X(s - \tau_1 a_1) \cap L_1(\tau_1) \cap L_2^- = X(s - \tau_1 a_1) \cap L_1(\tau_1) \ne \varnothing$, so, by Lemma 5.B.3, $(\tau_1, 0)$ is indeed a solution of (5.5.11).

Similarly, if $\xi_2 \le b_1$, then $B_0 \cap L_2^- \subseteq L_1^-$ and $(0, \tau_2)$ is a solution of (5.5.11).

iv. From now on, we can assume that $B_0 \cap L_1^- \cap \operatorname{int} L_2^+ \ne \varnothing$, $B_0 \cap L_2^- \cap \operatorname{int} L_1^+ \ne \varnothing$, where $\operatorname{int} L_i^+ := \{x \in \mathbb{E} : \langle a_i, x\rangle > b_i\}$, $i = 1, 2$. Combining this with (5.5.12), we obtain[7]

$$\operatorname{int} B_0 \cap L_1 \cap L_2^- \ne \varnothing, \qquad \operatorname{int} B_0 \cap L_2 \cap L_1^- \ne \varnothing. \tag{5.B.3}$$

Suppose $\langle a_2, H(s - \tau_1 a_1)\rangle \le b_2 \|s - \tau_1 a_1\|_*$ at Step 3. 1) If $s \ne \tau_1 a_1$, then $X(s - \tau_1 a_1)$ is a singleton, $x(s - \tau_1 a_1) = H(s - \tau_1 a_1)/\|s - \tau_1 a_1\|_*$, so we obtain $x(s - \tau_1 a_1) \in L_2^-$. Combining this with (5.B.2), we get $x(s - \tau_1 a_1) \in L_1(\tau_1) \cap L_2^-$. 2) If $s = \tau_1 a_1$, then $X(s - \tau_1 a_1) \cap L_1(\tau_1) \cap L_2^- = B_0 \cap L_1(\tau_1) \cap L_2^- \ne \varnothing$ in view of the first claim in (5.B.3) (recall that $L_1 \subseteq L_1(\tau_1)$). Thus, in any case, $X(s - \tau a_1) \cap L_1(\tau_1) \cap L_2^- \ne \varnothing$, and so, by Lemma 5.B.3, $(\tau_1, 0)$ is a solution of (5.5.11).

Similarly, one can consider the case when $\langle a_1, H(s - \tau_2 a_2)\rangle \le b_1 \|s - \tau_2 a_2\|_*$ at Step 3.

v. Suppose we have reached Step 4. From now on, we can assume that

$$\begin{aligned} &X(s - \tau_1 a_1) \cap L_1(\tau_1) \cap \operatorname{int} L_2^+ \ne \varnothing, \\ &X(s - \tau_2 a_2) \cap L_2(\tau_2) \cap \operatorname{int} L_1^+ \ne \varnothing. \end{aligned} \tag{5.B.4}$$

[7] Take an appropriate convex combination of two points from the specified nonempty convex sets.

Indeed, since both conditions at Step 3 have not been satisfied, $s \neq \tau_i a_i$, $i = 1, 2$, and $x(s - \tau_1 a_1) \notin L_2^-$, $x(s - \tau_2 a_2) \notin L_1^-$. Also, by (5.B.2), $x(s - \tau_i a_i) \in L_i(\tau_i)$, $i = 1, 2$.

Let $\mu \in \mathbb{R}^2_+$ be any solution of (5.5.11). By Lemma 5.B.3, $X(s - A\mu) \cap L_1(\mu_1) \cap L_2(\mu_2) \neq \varnothing$. Note that we cannot have $\mu_2 = 0$. Indeed, otherwise, we get $X(s - \mu_1 a_1) \cap L_1(\mu_1) \cap L_2^- \neq \varnothing$, so μ_1 must be a solution of (5.5.6) with parameters (s, a_1, b_1). But, by Lemma 5.B.2, such a solution is unique (in view of the second claim in (5.B.4)), $\langle a_1, x\rangle > b_1$ for some $x \in B_0$, so $b_1 < \|a_1\|_*$). Hence, $\mu_1 = \tau_1$, and we obtain a contradiction with (5.B.4). Similarly, we can show that $\mu_1 \neq 0$. Consequently, $\mu_1, \mu_2 > 0$, which means that μ is a solution of (5.5.8).

Thus, at this point, any solution of (5.5.11) must be a solution of (5.5.8). In view of Lemma 5.B.1, to finish the proof, it remains to show that a_1, a_2 are linearly independent and $\langle b, (A^*HA)^{-1} b\rangle < 1$. But this is simple. Indeed, from (5.B.4), it follows that

$$\text{either} \quad B_0 \cap L_1 \cap \operatorname{int} L_2^+ \neq \varnothing \quad \text{or} \quad B_0 \cap L_2 \cap \operatorname{int} L_1^+ \neq \varnothing \tag{5.B.5}$$

since τ_1 and τ_2 cannot both be equal to 0. Combining (5.B.5) and (5.B.3), we see that $\operatorname{int} B_0 \cap L_1 \cap L_2 \neq \varnothing$ and, in particular, $L_1 \cap L_2 \neq \varnothing$. Hence, a_1, a_2 are linearly independent (otherwise, $L_1 = L_2$, which contradicts (5.B.5)). Taking any $x \in \operatorname{int} B_0 \cap L_1 \cap L_2$, we obtain $\|x\| < 1$ and $A^* x = b$, hence $\langle b, (A^*HA)^{-1} b\rangle = \langle A^* x, (A^*HA)^{-1} A^* x\rangle \leq \|x\|^2 < 1$, where we have used $A(A^*HA)^{-1}A^* \preceq H^{-1}$. □

Chapter 6

Conclusions

6.1 Summary

In this thesis, we have presented several new results related to quasi-Newton methods.

First, we have studied classical quasi-Newton methods from the convex Broyden class and established certain efficiency estimates for their local superlinear convergence. One of the main conclusions of our analysis was that the BFGS method is almost insensitive to the condition number, in contrast to DFP. This is a nice theoretical confirmation of the well-known empirical superiority of BFGS over DFP.

Second, we have introduced a new family of greedy quasi-Newton methods. The most important feature of these methods, compared to the classical ones, is that they generate Hessian approximations converging to the exact Hessian. To achieve this, the greedy methods use a special greedy choice of the direction for updating Hessian approximations at each iteration. We have seen that the greedy methods are asymptotically faster than the classical ones but their superlinear convergence may start later than for the classical ones.

Finally, we have studied the Ellipsoid Method and realized that it has "incorrect" dependency on the dimensionality of the space. To address this problem, we have proposed a new variant of this algorithm—the Subgradient Ellipsoid Method. As we have seen, the efficiency estimate for this new method is nearly the best among the corresponding estimates for the Subgradient and Ellipsoid methods. In particular, the Subgradient Ellip-

soid Method withstands the passage to the limit when the dimensionality of the space tends to infinity, as does the Subgradient Method. We have also shown how to efficiently construct accuracy certificates in the Subgradient Ellipsoid Method, which is important for solving general problems with convex structure, such as saddle-point problems and variational inequalities.

6.2 Directions for Future Research

Let us outline some possible directions for further research.

The most natural research direction is, of course, obtaining *global efficiency estimates* for classical quasi-Newton methods. In this thesis, we have not addressed this question at all apart from the quadratic case.

Another interesting direction is the analysis of *limited-memory quasi-Newton methods*, such as L-BFGS [113], which are very popular for large-scale optimization.

One potential application for locally convergent quasi-Newton methods is to solve the subproblems arising in *path-following interior point methods*. By properly changing the penalty parameter in these methods, we can always make sure that the output of the previous subproblem is located in the region of local convergence of the new one. The local convergence results and the corresponding proof techniques, which we have presented in this thesis, could be very useful in this context.

Another idea is to use quasi-Newton methods for solving auxiliary subproblems arising in *high-order proximal-point* and *tensor methods* [135–137]. For example, the subproblem arising in the second-order tensor method (the Cubic Newton Method) is the minimization of a quadratic function regularized by the cube of the Euclidean norm; the subproblem arising in the second-order implementation of the third-order tensor method [138] is the minimization of a quadratic function regularized by the fourth power of the Euclidean norm. These subproblems have a very special structure which could be exploited in quasi-Newton methods (both classical and greedy).

Note that we have only considered the most basic problem formulation for quasi-Newton methods—smooth unconstrained optimization. However, in practice, we often need to solve more general problems, e.g, *composite optimization problems*, in which the objective is formed by the sum of two components: a smooth one and a general convex function with simple structure. It would be interesting to extend our results to this problem formulation.

One of the directions that we did not explore in this thesis, is the application of quasi-Newton methods to *nonsmooth problems*. A reasonable idea here might be to approximate a nonsmooth objective by a smooth one, add a small regularizer, and then apply, say, the standard BFGS for smooth optimization. In principle, we only need a little smoothing: even if the resulting approximation has an exponentially large condition number, this should not be a big problem for BFGS (at least, locally) since, as we showed in this thesis, the condition number enters the corresponding complexity bound under the logarithm.

Regarding the Subgradient Ellipsoid Method, it is not clear how to get rid of the extra logarithmic factor which appears whenever we want to use "more natural" time-varying step sizes instead of the constant ones. It would also be interesting to investigate whether this method can be *accelerated*, similarly to the Gradient Method on problems with Lipschitz continuous gradient.

Bibliography

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

[2] A. Alacaoglu, O. Fercoq, and V. Cevher. Random extrapolation for primal-dual coordinate descent. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 191–201. PMLR, 2020.

[3] A. Alacaoglu, Y. Malitsky, P. Mertikopoulos, and V. Cevher. A new regret analysis for Adam-type algorithms. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 202–210. PMLR, 2020.

[4] Z. Allen-Zhu, Z. Qu, P. Richtárik, and Y. Yuan. Even Faster Accelerated Coordinate Descent Using Non-Uniform Sampling. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 1110–1119. PMLR, 2016.

[5] Y. Arjevani and O. Shamir. Dimension-Free Iteration Complexity of Finite Sum Optimization Problems. *Advances in Neural Information Processing Systems*, 29, 2016.

[6] A. Auslender. Brève communication. Résolution numérique d'inégalités variationnelles. *R.A.I.R.O.*, 7(R2):67–72, 1973.

[7] M. Baes. Estimate Sequence Methods: Extensions and Approximations. IFOR Internal Report, ETH Zurich, 2009.

[8] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization*. Lecture notes, 2021.

[9] E. G. Birgin, J. Gardenghi, J. M. Martínez, S. A. Santos, and P. L. Toint. Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *Mathematical Programming*, 163:359–368, 2017.

[10] R. Bland, D. Goldfarb, and M. Todd. The Ellipsoid Method: A Survey. *Operations Research*, 29(6):1039–1091, 1981.

[11] P. T. Boggs and J. W. Tolle. Convergence Properties of a Class of Rank-two Updates. *SIAM J. Optim.*, 4(2):262–287, 1994.

[12] A. Bouaricha. Tensor Methods for Large, Sparse Unconstrained Optimization. *SIAM Journal on Optimization*, 7(3):732–756, 1997.

[13] N. Boumal. An introduction to optimization on smooth manifolds. To appear with Cambridge University Press. Mar. 2022.

[14] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[15] I. Brace and J. H. Manton. An improved BFGS-on-manifold algorithm for computing weighted low rank approximations. In *Proceedings of the 17th international Symposium on Mathematical Theory of Networks and Systems*, pages 1735–1738, 2006.

[16] C. G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19(92):577–593, 1965.

[17] C. G. Broyden. Quasi-Newton methods and their application to function minimization. *Mathematics of Computation*, 21(99):368–381, 1967.

[18] 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.

[19] C. G. Broyden. The convergence of a class of double-rank minimization algorithms: 2. The new algorithm. *IMA Journal of Applied Mathematics*, 6(3):222–231, 1970.

[20] C. G. Broyden, J. E. Dennis Jr, and J. Moré. On the local and superlinear convergence of quasi-Newton methods. *IMA Journal of Applied Mathematics*, 12(3):223–245, 1973.

[21] S. Bubeck and Y. T. Lee. Black-box Optimization with a Politician. In *International Conference on Machine Learning*, pages 1624–1631. PMLR, 2016.

[22] S. Bubeck, Y. T. Lee, and M. Singh. A geometric alternative to Nesterov's accelerated gradient descent. *arXiv preprint arXiv:1506.08187*, 2015.

[23] V. Bulatov and L. Shepot'ko. Method of centers of orthogonal simplexes for solving convex programming problems. *Methods of Optimization and Their Application*, 1982.

[24] R. Byrd and J. Nocedal. A tool for the analysis of quasi-Newton methods with application to unconstrained minimization. *SIAM Journal on Numerical Analysis*, 26(3):727–739, 1989.

[25] R. Byrd, J. Nocedal, and Y.-X. Yuan. Global convergence of a class of quasi-Newton methods on convex problems. *SIAM Journal on Numerical Analysis*, 24(5):1171–1190, 1987.

[26] R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer. A Stochastic Quasi-Newton Method for Large-Scale Optimization. *SIAM Journal on Optimization*, 26(2):1008–1031, 2016.

[27] R. H. Byrd, R. A. Tapia, and Y. Zhang. An SQP Augmented Lagrangian BFGS Algorithm for Constrained Optimization. *SIAM Journal on Optimization*, 2(2):210–241, 1992.

[28] C. Cartis, N. I. M. Gould, and P. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Mathematical Programming*, 127:245–295, 2011.

[29] C. Cartis, N. I. M. Gould, and P. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function-and derivative-evaluation complexity. *Mathematical Programming*, 130:295–319, 2011.

[30] C. Cartis, N. I. M. Gould, and P. L. Toint. Universal Regularization Methods: Varying the Power, the Smoothness and the Accuracy. *SIAM Journal on Optimization*, 29(1):595–615, 2019.

[31] A.-L. Cauchy. Méthode générale pour la résolution des systémes d'équations simultanées. *C. R. Acad. Sci. Paris*, 25:536–538, 1847.

[32] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27, 2011.

[33] T. F. Coleman and A. R. Conn. On the Local Convergence of a Quasi-Newton Method for the Nonlinear Programming Problem. *SIAM Journal on Numerical Analysis*, 21(4):755–769, 1984.

[34] A. R. Conn, N. I. M. Gould, and P. L. Toint. Convergence of quasi-Newton matrices generated by the symmetric rank one update. *Mathematical Programming*, 50:177–195, 1991.

[35] F. E. Curtis, T. Mitchell, and M. L. Overton. A BFGS-SQP method for nonsmooth, nonconvex, constrained optimization and its evaluation using relative minimization profiles. *Optimization Methods and Software*, 32(1):148–181, 2017.

[36] F. E. Curtis and M. L. Overton. A Sequential Quadratic Programming Algorithm for Nonconvex, Nonsmooth Constrained Optimization. *SIAM Journal on Optimization*, 22(2):474–500, 2012.

[37] F. E. Curtis and X. Que. A quasi-Newton algorithm for nonconvex, nonsmooth optimization with global convergence guarantees. *Mathematical Programming Computation*, 7:399–428, 2015.

[38] A. d'Aspremont. Smooth Optimization with Approximate Gradient. *SIAM Journal on Optimization*, 19(3):1171–1183, 2008.

[39] W. Davidon. Variable metric method for minimization. Technical report 5990, Argonne National Laboratory, 1959.

[40] W. Davidon. Variance algorithm for minimization. *Computer Journal*, 10(4):406–410, 1968.

[41] E. De Klerk, F. Glineur, and A. B. Taylor. Worst-Case Convergence Analysis of Inexact Gradient and Newton Methods Through Semidefinite Programming Performance Estimation. *SIAM Journal on Optimization*, 30(3):2053–2082, 2020.

[42] J. E. Dennis Jr. On Some Methods Based on Broyden's Secant Approximation to the Hessian. Technical Report 71-101, Cornell University, Ithaca, New York, 1971.

[43] J. E. Dennis Jr. On the convergence of Broyden's method for nonlinear systems of equations. *Mathematics of Computation*, 25(115):559–567, 1971.

[44] J. E. Dennis Jr. Toward a Unified Convergence Theory for Newton-Like Methods. In L. B. Rall, editor, *Nonlinear Functional Analysis and Applications*, pages 425–472. Academic Press, 1971.

[45] J. E. Dennis Jr and J. Moré. A characterization of superlinear convergence and its application to quasi-Newton methods. *Mathematics of Computation*, 28(126):549–560, 1974.

[46] J. E. Dennis Jr and J. Moré. Quasi-Newton Methods, Motivation and Theory. *SIAM Review*, 19(1):46–89, 1977.

[47] J. E. Dennis Jr and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, 1996.

[48] O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146:37–75, 2014.

[49] L. Dixon. Quasi Newton techniques generate identical points II: The proofs of four new theorems. *Mathematical Programming*, 3:345–358, 1972.

[50] L. Dixon. Quasi-Newton algorithms generate identical points. *Mathematical Programming*, 2:383–387, 1972.

[51] N. Doikov. *New Second-Order and Tensor Methods in Convex Optimization*. PhD thesis, Université catholique de Louvain (UCL), 2021.

[52] N. Doikov and Y. Nesterov. Inexact Tensor Methods with Dynamic Accuracies. In *Proceedings of the 37th International Conference on Machine Learning*, pages 2577–2586, 2020.

[53] N. Doikov and Y. Nesterov. Local convergence of tensor methods. *Mathematical Programming*:1–22, 2021.

[54] N. Doikov and Y. Nesterov. Minimizing Uniformly Convex Functions by Cubic Regularization of Newton Method. *Journal of Optimization Theory and Applications*, 189:317–339, 2021.

[55] Y. Drori and M. Teboulle. Performance of first-order methods for smooth convex minimization: a novel approach. *Mathematical Programming*, 145:451–482, 2014.

[56] J. Duchi, E. Hazan, and Y. Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12(7), 2011.

[57] P. Dvurechensky and A. Gasnikov. Stochastic Intermediate Gradient Method for Convex Problems with Stochastic Inexact Oracle. *Journal of Optimization Theory and Applications*, 171:121–145, 2016.

[58] O. Fercoq and P. Richtárik. Accelerated, Parallel, and Proximal Coordinate Descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.

[59] R. Fletcher. A new approach to variable metric algorithms. *Computer Journal*, 13(3):317–322, 1970.

[60] R. Fletcher. An Optimal Positive Definite Update for Sparse Hessian Matrices. *SIAM Journal on Optimization*, 5(1):192–218, 1995.

[61] R. Fletcher. *Practical Methods of Optimization*. Wiley, 2000.

[62] R. Fletcher and M. J. D. Powell. A rapidly convergent descent method for minimization. *Computer Journal*, 6(2):163–168, 1963.

[63] D. Gabay. Minimizing a differentiable function over a differential manifold. *Journal of Optimization Theory and Applications*, 37(2):177–219, 1982.

[64] U. M. Garcia-Palomares and O. L. Mangasarian. Superlinearly convergent quasi-Newton algorithms for nonlinearly constrained optimization problems. *Mathematical Programming*, 11:1–13, 1976.

[65] A. Gasnikov, P. Dvurechensky, E. Gorbunov, E. Vorontsova, D. Selikhanovych, C. A. Uribe, B. Jiang, H. Wang, S. Zhang, S. Bubeck, Q. Jiang, Y. T. Lee, Y. Li, and A. Sidford. Near Optimal Methods for Minimizing Convex Functions with Lipschitz p-th Derivatives. In *Conference on Learning Theory*, pages 1392–1393. PMLR, 2019.

[66] R. Ge and M. J. D. Powell. The convergence of variable metric matrices in unconstrained optimization. *Math. Program.*, 27:123–143, 1983.

[67] S. Ghadimi and G. Lan. Optimal Stochastic Approximation Algorithms for Strongly Convex Stochastic Composite Optimization I: A Generic Algorithmic Framework. *SIAM Journal on Optimization*, 22(4):1469–1492, 2012.

[68] S. Ghadimi and G. Lan. Optimal Stochastic Approximation Algorithms for Strongly Convex Stochastic Composite Optimization, II: Shrinking Procedures and Optimal Algorithms. *SIAM Journal on Optimization*, 23(4):2061–2089, 2013.

[69] D. Goldfarb. Sufficient conditions for the convergence of a variable metric algorithm. In R. Fletcher, editor, *Optimization*, pages 273–281, London / New York. Academic Press, 1969.

[70] D. Goldfarb. A Family of Variable-Metric Methods Derived by Variational Means. *Mathematics of Computation*, 24(109):23–26, 1970.

[71] D. Goldfarb, Y. Ren, and A. Bahamou. Practical Quasi-Newton Methods for Training Deep Neural Networks. *Advances in Neural Information Processing Systems*, 33:2386–2396, 2020.

[72] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 4th edition, 2013.

[73] R. Gower, D. Goldfarb, and P. Richtárik. Stochastic Block BFGS: Squeezing More Curvature out of Data. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1869–1878. PMLR, 2016.

[74] R. M. Gower and P. Richtárik. Randomized Quasi-Newton Updates are Linearly Convergent Matrix Inversion Algorithms. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1380–1409, 2017.

[75] R. M. Gower, M. Schmidt, F. Bach, and P. Richtárik. Variance-Reduced Methods for Machine Learning. *Proceedings of the IEEE*, 108(11):1968–1983, 2020.

[76] G. N. Grapiglia and Y. Nesterov. Regularized Newton Methods for Minimizing Functions with Hölder Continuous Hessians. *SIAM Journal on Optimization*, 27(1):478–506, 2017.

[77] G. N. Grapiglia and Y. Nesterov. Accelerated Regularized Newton Methods for Minimizing Composite Convex Functions. *SIAM Journal on Optimization*, 29(1):77–99, 2019.

[78] G. N. Grapiglia and Y. Nesterov. Tensor methods for finding approximate stationary points of convex functions. *Optimization Methods and Software*:1–34, 2020.

[79] G. N. Grapiglia and Y. Nesterov. Tensor Methods for Minimizing Convex Functions with Hölder Continuous Higher-Order Derivatives. *SIAM Journal on Optimization*, 30(4):2750–2779, 2020.

[80] G. N. Grapiglia and Y. Nesterov. On inexact solution of auxiliary problems in tensor methods for convex optimization. *Optimization Methods and Software*, 36(1):145–170, 2021.

[81] J. Greenstadt. Variations on variable-metric methods. *Mathematics of Computation*, 24(109):1–22, 1970.

[82] A. Griewank and P. L. Toint. On the unconstrained optimization of partially separable functions. In M. J. D. Powell, editor, *Nonlinear Optimization*, pages 301–312. Academic Press, London, 1981.

[83] A. Griewank and P. L. Toint. Local convergence analysis for partitioned quasi-Newton updates. *Numerische Mathematik*, 39(3):429–448, 1982.

[84] A. Griewank and P. L. Toint. Partitioned variable metric updates for large structured optimization problems. *Numerische Mathematik*, 39:119–137, 1982.

[85] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[86] J. Guo. *Smooth Quasi-Newton Methods for Nonsmooth Optimization.* PhD thesis, Cornell University, 2018.

[87] S.-P. Han. Superlinearly convergent variable metric algorithms for general nonlinear programming problems. *Mathematical Programming*, 11:263–282, 1976.

[88] S.-P. Han. A globally convergent method for nonlinear programming. *Journal of Optimization Theory and Applications*, 22(3):297–309, 1977.

[89] S.-P. Han. Dual Variable Metric Algorithms for Constrained Optimization. *SIAM Journal on Control and Optimization*, 15(4):546–565, 1977.

[90] J. Hu, B. Jiang, L. Lin, Z. Wen, and Y.-X. Yuan. Structured Quasi-Newton Methods for Optimization with Orthogonality Constraints. *SIAM Journal on Scientific Computing*, 41(4):A2239–A2269, 2019.

[91] W. Huang, P.-A. Absil, and K. A. Gallivan. A Riemannian BFGS Method Without Differentiated Retraction for Nonconvex Optimization Problems. *SIAM Journal on Optimization*, 28(1):470–495, 2018.

[92] W. Huang, K. A. Gallivan, and P.-A. Absil. A Broyden Class of Quasi-Newton Methods for Riemannian Optimization. *SIAM Journal on Optimization*, 25(3):1660–1685, 2015.

[93] R. Johnson and T. Zhang. Accelerating Stochastic Gradient Descent using Predictive Variance Reduction. *Advances in Neural Information Processing Systems*, 26, 2013.

[94] D. Kamzolov, P. Dvurechensky, and A. Gasnikov. Universal intermediate gradient method for convex problems with inexact oracle. *Optimization Methods and Software*:1–28, 2020.

[95] N. Karmarkar. A New Polynomial-Time Algorithm for Linear Programming. *Combinatorica*, 4(4):373–395, 1984.

[96] A. Kavis, K. Y. Levy, F. Bach, and V. Cevher. UniXGrad: A Universal, Adaptive Algorithm with Optimal Guarantees for Constrained Optimization. In *Advances in Neural Information Processing Systems*, volume 32, pages 6257–6266, 2019.

[97] L. Khachiyan. A polynomial algorithm in linear programming. In *Soviet Mathematics Doklady*, volume 244 of number 5, pages 1093–1096. Russian Academy of Sciences, 1979.

[98] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[99] A. Kostrikin and Y. Manin. *Linear Algebra and Geometry*. Gordon and Breach Science Publishers, 1989.

[100] D. Kovalev, R. M. Gower, P. Richtárik, and A. Rogozin. Fast linear convergence of randomized BFGS. *arXiv preprint arXiv:2002.11337*, 2020.

[101] G. Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133:365–397, 2012.

[102] G. Lan. *First-order and Stochastic Optimization Methods for Machine Learning*. Springer, 2020.

[103] G. Lan, A. Nemirovski, and A. Shapiro. Validation analysis of mirror descent stochastic approximation method. *Mathematical Programming*, 134:425–458, 2012.

[104] G. Lan and Y. Zhou. An optimal randomized incremental gradient method. *Mathematical Programming*, 171:167–215, 2018.

[105] N. Le Roux, M. Schmidt, and F. Bach. A Stochastic Gradient Method with an Exponential Convergence Rate for Finite Training Sets. *Advances in Neural Information Processing Systems*, 25, 2012.

[106] Y. T. Lee and A. Sidford. Efficient Accelerated Coordinate Descent Methods and Faster Algorithms for Solving Linear Systems. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 147–156. IEEE, 2013.

[107] A. Levin. An algorithm for minimizing convex functions. In *Doklady Akademii Nauk SSSR*, volume 160 of number 6, pages 1244–1247. Russian Academy of Sciences, 1965.

[108] K. Y. Levy, A. Yurtsever, and V. Cevher. Online Adaptive Methods, Universality and Acceleration. *Advances in Neural Information Processing Systems*, 31, 2018.

[109] A. Lewis and M. Overton. Nonsmooth optimization via quasi-Newton methods. *Mathematical Programming*, 141:135–163, 2013.

[110] D. Lin, H. Ye, and Z. Zhang. Explicit Convergence Rates of Greedy and Random Quasi-Newton Methods. *Journal of Machine Learning Research*, 23(162):1–40, 2022.

[111] Q. Lin, Z. Lu, and L. Xiao. An Accelerated Proximal Coordinate Gradient Method. *Advances in Neural Information Processing Systems*, 27, 2014.

[112] Q. Lin, Z. Lu, and L. Xiao. An Accelerated Randomized Proximal Coordinate Gradient Method and its Application to Regularized Empirical Risk Minimization. *SIAM Journal on Optimization*, 25(4):2244–2273, 2015.

[113] D. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

[114] Z. Lu and L. Xiao. On the complexity analysis of randomized block-coordinate descent methods. *Mathematical Programming*, 152:615–642, 2015.

[115] J. M. Martínez. On High-Order Model Regularization for Constrained Optimization. *SIAM Journal on Optimization*, 27(4):2447–2458, 2017.

[116] A. Mokhtari, M. Eisen, and A. Ribeiro. IQN: An Incremental Quasi-Newton Method with Local Superlinear Convergence Rate. *SIAM Journal on Optimization*, 28(2):1670–1698, 2018.

[117] A. Mokhtari and A. Ribeiro. RES: Regularized Stochastic BFGS Algorithm. *IEEE Transactions on Signal Processing*, 62(23):6089–6104, 2014.

[118] R. D. Monteiro and B. F. Svaiter. An Accelerated Hybrid Proximal Extragradient Method for Convex Optimization and its Implications to Second-Order Methods. *SIAM Journal on Optimization*, 23(2):1092–1125, 2013.

[119] P. Moritz, R. Nishihara, and M. Jordan. A Linearly-Convergent Stochastic L-BFGS Algorithm. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 249–258. PMLR, 2016.

[120] I. Necoara, Y. Nesterov, and F. Glineur. Random Block Coordinate Descent Methods for Linearly Constrained Optimization over Networks. *Journal of Optimization Theory and Applications*, 173:227–254, 2017.

[121] I. Necoara, A. Patrascu, and F. Glineur. Complexity of first-order inexact Lagrangian and penalty methods for conic convex programming. *Optimization Methods and Software*, 34(2):305–335, 2019.

[122] A. Nemirovski. *Information-Based Complexity of Convex Programming*. Lecture notes, 1995.

[123] A. Nemirovski. Prox-Method with Rate of Convergence $O(1/t)$ for Variational Inequalities with Lipschitz Continuous Monotone Operators and Smooth Convex-Concave Saddle Point Problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

[124] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust Stochastic Approximation Approach to Stochastic Programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

[125] A. Nemirovski, S. Onn, and U. G. Rothblum. Accuracy Certificates for Computational Problems with Convex Structure. *Mathematics of Operations Research*, 35(1):52–78, 2010.

[126] A. Nemirovsky and D. Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience, 1983.

[127] 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.

[128] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103:127–152, 2005.

[129] Y. Nesterov. Accelerating the cubic regularization of Newton's method on convex problems. *Mathematical Programming*, 112:159–181, 2008.

[130] Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120:221–259, 2009.

[131] Y. Nesterov. Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[132] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140:125–161, 2013.

[133] Y. Nesterov. *Lectures on Convex Optimization*, volume 137. Springer, 2018.

[134] Y. Nesterov. Inexact basic tensor methods for some classes of convex optimization problems. *Optimization Methods and Software*:1–29, 2020.

[135] Y. Nesterov. Implementable tensor methods in unconstrained convex optimization. *Mathematical Programming*, 186:157–183, 2021.

[136] Y. Nesterov. Inexact accelerated high-order proximal-point methods. *Mathematical Programming*, 2021.

[137] Y. Nesterov. Inexact High-Order Proximal-Point Methods with Auxiliary Search Procedure. *SIAM Journal on Optimization*:2807–2828, 2021.

[138] Y. Nesterov. Superfast Second-Order Methods for Unconstrained Convex Optimization. *Journal of Optimization Theory and Applications*, 191:1–30, 2021.

[139] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, 1994.

[140] Y. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108:177–205, 2006.

[141] Y. Nesterov and S. U. Stich. Efficiency of the Accelerated Coordinate Descent Method on Structured Optimization Problems. *SIAM Journal on Optimization*, 27(1):110–123, 2017.

[142] D. Newman. Location of the maximum on unimodal surfaces. *Journal of the ACM*, 12(3):395–398, 1965.

[143] J. Nocedal and M. L. Overton. Projected Hessian Updating Algorithms for Nonlinearly Constrained Optimization. *SIAM Journal on Numerical Analysis*, 22(5):821–850, 1985.

[144] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Science & Business Media, 2006.

[145] J. M. Papakonstantinou. *Historical Development of the BFGS Secant Method and Its Characterization Properties*. PhD thesis, Rice University, 2009.

[146] D. Perekrestenko, V. Cevher, and M. Jaggi. Faster Coordinate Descent via Adaptive Importance Sampling. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, pages 869–877. PMLR, 2017.

[147] A. Perry. A Class of Conjugate Gradient Algorithms with a Two-Step Variable Metric Memory. Discussion Paper 269, Northwestern University, Evanston, Illinois, 1977.

[148] M. J. D. Powell. A New Algorithm for Unconstrained Optimization. In *Nonlinear Programming*, pages 31–65. Elsevier, 1970.

[149] M. J. D. Powell. On the convergence of the variable metric algorithm. *IMA Journal of Applied Mathematics*, 7(1):21–36, 1971.

[150] M. J. D. Powell. Some global convergence properties of a variable metric algorithm for minimization without exact line searches. In R. W. Cottle and C. E. Lemke, editors, *Nonlinear Programming, SIAM-AMS proceedings*, volume 9. American Mathematical Society, 1976.

[151] M. J. D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. In *Numerical Analysis*, pages 144–157. Springer, 1977.

[152] M. J. D. Powell. Algorithms for nonlinear constraints that use Lagrangian functions. *Mathematical Programming*, 14:224–248, 1978.

[153] B. N. Pshenichny and Y. M. Danilin. *Numerical Methods in Extremal Problems*. Mir Publishers, 1978.

[154] C. Qi, K. A. Gallivan, and P.-A. Absil. Riemannian BFGS Algorithm with Applications. In *Recent Advances in Optimization and its Applications in Engineering*, pages 183–192. Springer, 2010.

[155] Z. Qu, P. Richtárik, M. Takác, and O. Fercoq. SDNA: Stochastic Dual Newton Ascent for Empirical Risk Minimization. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 1823–1832. PMLR, 2016.

[156] J. Renegar. A polynomial-time algorithm, based on Newton's method, for linear programming. *Mathematical Programming*, 40:59–93, 1988.

[157] P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144:1–38, 2014.

[158] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[159] A. Rodomanov and Y. Nesterov. Greedy Quasi-Newton Methods with Explicit Superlinear Convergence. *SIAM Journal on Optimization*, 31(1):785–811, 2021.

[160] A. Rodomanov and Y. Nesterov. New Results on Superlinear Convergence of Classical Quasi-Newton Methods. *Journal of Optimization Theory and Applications*, 188:744–769, 2021.

[161] A. Rodomanov and Y. Nesterov. Rates of superlinear convergence for classical quasi-Newton methods. *Mathematical Programming*, 194:159–190, 2022.

[162] A. Rodomanov and Y. Nesterov. Subgradient ellipsoid method for nonsmooth convex problems. *Mathematical Programming*, 2022.

[163] B. Savas and L.-H. Lim. Quasi-Newton Methods on Grassmannians and Multilinear Approximations of Tensors. *SIAM Journal on Scientific Computing*, 32(6):3352–3393, 2010.

[164] M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162:83–112, 2017.

[165] R. B. Schnabel and T.-T. Chow. Tensor Methods for Unconstrained Optimization Using Second Derivatives. *SIAM Journal on Optimization*, 1(3):293–315, 1991.

[166] N. N. Schraudolph, J. Yu, and S. Günter. A Stochastic Quasi-Newton Method for Online Convex Optimization. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2, pages 436–443. PMLR, 2007.

[167] L. Schubert. Modification of a quasi-Newton method for nonlinear equations with a sparse Jacobian. *Mathematics of Computation*, 24(109):27–30, 1970.

[168] S. Shalev-Shwartz and T. Zhang. Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization. *Journal of Machine Learning Research*, 14(2), 2013.

[169] D. Shanno. Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation*, 24(111):647–656, 1970.

[170] D. Shanno. Conjugate Gradient Methods with Inexact Searches. *Mathematics of Operations Research*, 3(3):244–256, 1978.

[171] D. Shanno. On variable-metric methods for sparse Hessians. *Mathematics of Computation*, 34(150):499–514, 1980.

[172] N. Z. Shor. Convergence rate of the gradient descent method with dilatation of the space. *Cybernetics*, 6(2):102–108, 1970.

[173] N. Z. Shor. Cut-off method with space extension in convex programming problems. *Cybernetics*, 13(1):94–96, 1977.

[174] J. Sohl-Dickstein, B. Poole, and S. Ganguli. Fast large-scale optimization by unifying stochastic gradient and quasi-Newton methods. In *Proceedings of the 31st International Conference on Machine Learning*, pages 604–612. PMLR, 2014.

[175] J. Stoer. The convergence of matrices generated by rank-2 methods from the restricted β-class of Broyden. *Numer. Math.*, 44:37–52, 1984.

[176] F. Stonyakin, A. Tyurin, A. Gasnikov, P. Dvurechensky, A. Agafonov, D. Dvinskikh, M. Alkousa, D. Pasechnyuk, S. Artamonov, and V. Piskunova. Inexact model: a framework for optimization and variational inequalities. *Optimization Methods and Software*:1–47, 2021.

[177] R. A. Tapia. Diagonalized multiplier methods and quasi-Newton methods for constrained optimization. *Journal of Optimization Theory and Applications*, 22(2):135–194, 1977.

[178] R. A. Tapia. On secant updates for use in general constrained optimization. *Mathematics of Computation*, 51(183):181–202, 1988.

[179] S. Tarasov, L. Khachiyan, and I. Erlikh. The method of inscribed ellipsoids. In *Soviet Mathematics Doklady*, volume 37 of number 1, pages 226–230, 1988.

[180] A. B. Taylor, J. M. Hendrickx, and F. Glineur. Exact Worst-Case Performance of First-Order Methods for Composite Convex Optimization. *SIAM Journal on Optimization*, 27(3):1283–1313, 2017.

[181] A. B. Taylor, J. M. Hendrickx, and F. Glineur. Performance estimation toolbox (PESTO): Automated worst-case analysis of first-order optimization methods. In *2017 IEEE 56th Annual Conference on Decision and Control*, pages 1278–1283. IEEE, 2017.

[182] A. B. Taylor, J. M. Hendrickx, and F. Glineur. Smooth strongly convex interpolation and exact worst-case performance of first-order methods. *Mathematical Programming*, 161:307–345, 2017.

[183] A. B. Taylor, J. M. Hendrickx, and F. Glineur. Exact Worst-Case Convergence Rates of the Proximal Gradient Method for Composite Convex Minimization. *Journal of Optimization Theory and Applications*, 178:455–476, 2018.

[184] P. L. Toint. On sparse and symmetric matrix updating subject to a linear equation. *Mathematics of Computation*, 31(140):954–961, 1977.

[185] P. L. Toint. Some numerical results using a sparse matrix updating formula in unconstrained optimization. *Mathematics of Computation*, 32(143):839–851, 1978.

[186] P. L. Toint. A note about sparsity exploiting quasi-Newton updates. *Mathematical Programming*, 21:172–181, 1981.

[187] P. L. Toint. A sparse quasi-Newton update derived variationally with a nondiagonally weighted Frobenius norm. *Mathematics of Computation*, 37(156):425–433, 1981.

[188] P. L. Toint. Towards an Efficient Sparsity Exploiting Newton Method for Minimization. In I. S. Duff, editor, *Sparse Matrices and Their Uses*, pages 57–88. Academic Press, London, England, 1981.

[189] Q. Tran-Dinh, O. Fercoq, and V. Cevher. A Smooth Primal-Dual Optimization Framework for Nonsmooth Composite Convex Minimization. *SIAM Journal on Optimization*, 28(1):96–134, 2018.

[190] X. Wang, S. Ma, D. Goldfarb, and W. Liu. Stochastic Quasi-Newton Methods for Nonconvex Stochastic Optimization. *SIAM Journal on Optimization*, 27(2):927–956, 2017.

[191] B. E. Woodworth and N. Srebro. Tight Complexity Bounds for Optimizing Composite Objectives. *Advances in Neural Information Processing Systems*, 29, 2016.

[192] J. Yu, S. Vishwanathan, S. Günter, and N. N. Schraudolph. A Quasi-Newton Approach to Nonsmooth Convex Optimization Problems in Machine Learning. *The Journal of Machine Learning Research*, 11:1145–1200, 2010.

[193] D. Yudin and A. Nemirovskii. Informational complexity and efficient methods for the solution of convex extremal problems. *Matekon*, 13(2):22–45, 1976.