Institute of Information and Communication Technologies,
Electronics and Applied Mathematics

Convex Interpolation and Performance Estimation of First-order Methods for Convex Optimization

Adrien Taylor

Thesis submitted in partial fulfillment of the requirements for the degree of
Docteur en sciences de l'ingénieur

Dissertation committee:
Coralia Cartis (Oxford University)
Etienne de Klerk (Tilburg University and Delft University of Technology)
François Glineur (Université catholique de Louvain, advisor)
Julien Hendrickx (Université catholique de Louvain, advisor)
Philippe Lefèvre (Université catholique de Louvain, chairman)
Yurii Nesterov (Université catholique de Louvain)

Abstract

The goal of this thesis is to show how to derive in a completely automated way exact and global worst-case guarantees for first-order methods in convex optimization. To this end, we formulate a generic optimization problem looking for the worst-case scenarios.

The worst-case computation problems, referred to as *performance estimation problems* (PEPs), are intrinsically infinite-dimensional optimization problems formulated over a given class of objective functions. To render those problems tractable, we develop (smooth and non-smooth) *convex interpolation* framework, which provides necessary and sufficient conditions to interpolate our objective functions.

With this idea, we transform PEPs into solvable finite-dimensional semidefinite programs, from which one obtains worst-case guarantees and worst-case functions, along with the corresponding explicit proofs.

PEPs already proved themselves very useful as a tool for developing convergence analyses of first-order optimization methods. Among others, PEPs allow obtaining exact guarantees for gradient methods, along with their inexact, projected, proximal, conditional, decentralized and accelerated versions.

Acknowledgements

A thesis is a long journey that can hardly be completed without support. I was very lucky for the work and personal environment I have been living in during those last years, and I wish to thank the many persons who contributed to it.

First of all, I would like to thank my two advisers, François and Julien, for giving me the opportunity to work within the mathematical engineering department, for their support and guidances. Also, I am very grateful to Moritz Diehl, Etienne de Klerk, Coralia Cartis, Yurii Nesterov and Philippe Lefèvre for their insightful feedbacks, and for the time they spent for me. In addition, I am greatly indebted to other researchers I have met during those last years, for all the scientific discussions, motivations and advise. Among others, big thanks to Matthew Philippe, Romain Hollanders, François Gonze, Pierre-Yves Chevalier, Andrea Simonetto, Nicolas Boumal, Ion Necoara and Frank Iutzeler.

I wish to thank the colleagues and friends with whom I had the pleasure to spend time at the office. Among the many others, my two former officemates Pyc and François certainly deserve a particular attention for bearing my (often unstoppable) long speeches. Also, I could not emphasize enough the importance of the technical and administrative staff of the department and of all the time they spend making the Euler so enjoyable: thanks to Isabelle, Marie-Christine, Nathalie, Carine, Etienne, François and Ludovic.

Those last years, I also had the chance to live in those great places that were *le Petit-Ry* and *la rue Philippe Baucq*, with passionate persons sharing similar convictions, ways, jobs, and hobbies; for that, huge thanks to Bouny, Antoinette, Alex, Ade, Bob, Anais, Laulau, Tom, Coco, Juju (J.), Oli, Théthé, Pyf, Ju (C.), Ol., Delphine, Vio, Adri (D.), Stéphanie and Eliane.

I would also like to thank different groups of persons who contributed in one way or another to the accomplishment of this work, through support, drinks, sport, music, board games, robotic design, holidays, debates, or simply by enjoying good time together. Among them, I wish to particularly thank Adri (C.), Gauthier, Nono, Ludi, Thomas (R.), Bigno, Coin, Emilie, Carl, Barnab, Aileen, Marie, Mehdi, Roxane, Ben and Thomas (G.) for their constant enthusiasm, smiles and precious advise. Finally, I could not finish those lines without thanking the old-timers Gi, Jeje and Domi for their caring presence along the years.

Pour finir, un énorme merci à Emily et Jojo, pour leur patience, leurs attentions, et pour le fait de toujours parvenir à supporter après toutes ces années. Un grand merci également à mes parents Sabine et Olivier, ainsi qu'à leurs compagnons respectifs Pascal et Maria-Paola, pour leurs oreilles attentives, leurs conseils et leur bienveillance.

Contents

New answers to simple questions

This section is intended for the informed readers: we present a few spoilers on (perhaps surprising) results taken from the following chapters, expressed in their simplest forms. In order to do that, let f be a smooth (strongly) convex function, and consider the corresponding unconstrained minimization problem

$$\min_{x \in \mathbb{R}^d} f(x).$$

Given an iterative algorithm and an initial point x_0, the performance estimation problem that is the main focus of this thesis consists in identifying the functions achieving the worst value of objective function accuracy after N iterations, i.e. that maximize $f(x_N) - f(x_*)$.

What are the worst-case functions for the fixed-step gradient, fast gradient and optimized gradient methods in smooth convex unconstrained minimization ?

One-dimensional Huber losses and quadratics. (**Chapter 4**)

For smooth strongly convex unconstrained minimization with the gradient method, is there a benefit of using exact line search as compared to fixed step sizes ?

With an appropriate step size, the worst-cases are the same. (**Chapter 6**)

In smooth strongly convex unconstrained minimization, how does the worst-case guarantee change when inaccurate search directions are tolerated within an exact line search steepest descent scheme ?

Linear convergence is preserved, albeit with a worsened condition number. (**Chapter 6**)

When analyzing the global worst-case guarantee of *simple* first-order methods, is there an upper bound on the number of (in)equalities required in the proof ?

Yes, we only need to consider interpolation conditions (from the class of problems under consideration). The number of such conditions required to write a *tight* convergence proof depends on the number of iterations, and on the class of functions of interest. It is a priori $\mathcal{O}(N^2)$ for non-smooth and smooth (strongly) convex problems. (**Chapter 4** and **Chapter 5**)

Chapter 1

Introduction

1.1 What is optimization ?

Mathematical optimization is the science of making decisions in a rational way. It drives many of today's theories in physics (for example least-action principles dictating equations of motions, or physical systems going to their states of minimal energy), in biology (Darwin's theory of evolution favoring survival of individuals with most beneficial traits) or even in economy (individual behaviors are predicted by assuming they maximize their utilities).

Due to the rise of larger computational capabilities, optimization has also become a very powerful tool for making decisions in practice. Formally, the aim of mathematical optimization is to chose some parameter x within a set X such that $f(x)$ is as small as possible:

$$\min_{x \in X} f(x). \tag{OPT}$$

Writing an optimization problem in this form is referred to as *modelling*. A lot of practical applications can be modelled and are actually being solved using this formalism — we refer to [RM09, SNW12, PE10, BV04] for examples of uses in automatic control, machine learning, signal processing, inverse problems, statistical modelling and other engineering applications.

Although very general, the modelling process should never be underestimated. In fact, almost every problem in our daily life can be modelled in the optimization framework. However, there are relatively few instances that we can actually *solve* in practice, as there exists no *universal* way to solve (OPT) within a reasonable time. That is, even by adding some assumptions (e.g., differentiability), it is generally possible to design problems of the form (OPT) such that the computational cost for solving them is just too high to be even

considered. Therefore, it is commonly admitted that a good modelling of the problem is usually the key for solving it, even approximately.

As there is no guarantee for solving (OPT) in general, optimizers commonly add assumptions on both X and f in order to design algorithms with working certifications. Those assumptions should be motivated in two aspects: on the one hand they should allow for designing efficient algorithms for solving the corresponding (OPT), and on the other hand, they should contain a sufficiently rich class of problems to embed as many applications as possible — those are of course conflicting goals, but both aspects are nevertheless essential.

In what follows, we focus on continuous optimization problems, that is, when variables are allowed to take their values within a continuum. More precisely, we mostly focus on the case where X is a convex set, and f a convex function. Those assumptions are motivated on the one hand by a variety of applications (see e.g., [PE10, BV04] and the references therein), and on the other hand by strong theoretical guarantees. Also, we focus on the class of first-order optimization algorithms, which proved themselves particularly successful in the context of *large-scale* decision making, that is, when the decision space X is high-dimensional (more details follow in the next sections).

The goal of this thesis is to provide a framework for automatically analyzing the worst-case performances of first-order methods in convex optimization. We devote the next section to briefly describe the broader context in which those analyses take place.

1.2 Recent trends in large-scale optimization

Since the beginning of the modern computational optimization era, starting after the second world war with among others the work of Dantzig on linear programming and the simplex method [Dan98], optimization has undergone several paradigm shifts. This resulted in a vast literature on large families of algorithms tailored for particular classes and instances of (OPT).

In the context of continuous optimization, two families of algorithms are particularly used: methods using first-order derivatives (gradients) on the one hand, and methods using second-order derivatives (Hessian) on the other one. In the sequel, we only treat first-order methods; however, depending on the specific application, it may be much more appropriate to use second-order schemes.

- ◇ Second-order (or Newton) methods are usually the algorithms of choice for obtaining highly-accurate solutions (see e.g., [NW06, BV04]). However, this usually comes at the price of large iteration costs.

  Numerous numerical schemes were developed for trying to alleviate the computational burden coming with second-order information, while keeping their accuracy. For example, it is common to use approximate second-

order information or to exploit specific problem structures (e.g., [NW06, Gon12b, FG16]). In convex optimization, second-order schemes are particularly studied for interior-point methods (see e.g., [Gon12a, Ren01, NN94]).

- ◇ On the other hand, much attention is currently being given to first-order methods. Those methods have an old history, but were usually not viewed as methods of choice, especially due to their difficulties in obtaining very accurate solutions. Contrasting with second-order methods, first-order methods generally benefits from a much cheaper cost per iteration.

  First-order methods are prominent in today's computational optimization practices. This is due to their low iteration cost, but also to a quantity of practical large-scale problems that do not actually require accurate solutions. For example, a lot of objective functions arising in machine learning, signal and image processing problems can seen as approximations (see e.g., [BB08]). For those problems, it is unnecessary to obtain solutions which are more accurate than the objectives themselves.

The previous points motivated a lot of different research tracks for rendering large-scale computations computationally feasible and as light as possible. In that direction, much attention is currently being given to *structural optimization* (we borrow this term from [Nes04, Nes08]), whose focus is to exploit the particular structure of common optimization problems for designing efficient tailored methods. In that context, very common examples include:

- ◇ block-coordinate descent schemes which treat a subset of the variables at a time in order to maintain cheap iteration costs (see e.g. [RT14, FR15, Nes12a]). Those methods are particularly suited for problems with low interactions between blocks of variables.
- ◇ Incremental or stochastic gradient descent schemes, which use partial knowledge on the objective function at each iteration (see e.g., [Ber10, SSBD14, Kiw04] and [RM51] for the original presentation). Those methods are designed for objective functions written as averages of simple functions.
- ◇ Distributed and/or decentralized methods, which use particular structures in order to split the optimization process among different computational units, either in order to lighten the computational burden of a single unit, or because one can not rely on a central computational unit (see e.g., [BPC+11, NO09]).
- ◇ Numerous other techniques exist for improving either the convergence rates and/or the cost per iteration. As examples, we cite smoothing techniques (see e.g., [DGN12, BT12, Nes05]) whose aim is to improve the convergence rates of first-order schemes on well-structured non-smooth problems; and sparsity-exploiting methods that may render iteration cost sublinear in the dimension of the problem (see e.g., [Nes14]).

1.3 Worst-case analyses of numerical schemes

The main contributions of this work take place in the context of worst-case analyses of first-order optimization algorithms. Our goal is to obtain global guarantees on various measures of accuracy achieved by optimization methods, for a given computational cost. Those guarantees are essential as they allow comparing and choosing the most appropriate methods for solving (OPT).

Comparing the efficiency of optimization algorithms can be carried out in a variety of different manners. In this work, we focus on the concepts of *absolute inaccuracy* and *efficiency* (or *worst-case inaccuracy*). For that purpose, we consider three ingredients:

- ◇ a class of problems instances $\mathcal{P}$ containing the problems of interest (f, X).
- ◇ A method M that produces an approximate solution $x_{M(f,X)}$ when it is given a problem instance $(f, X) \in \mathcal{P}$ (i.e., M has a built-in stopping criterion).
- ◇ A performance measure, typically $f(x_{M(f,X)}) - f(x_*)$ with

$$x_* = \operatorname*{argmin}_{x \in X} f(x)$$

  (for other performance measures, see Chapter 4 and Chapter 5).

From those elements, we define the *absolute inaccuracy* of method M on an instance (f, X) as the value of the performance measure evaluated at the output of M:

$$\varepsilon(M(f, X), f, X) = f(x_{M(f,X)}) - f(x_*).$$

In the same way, the *efficiency* of M is defined as its worst-case absolute inaccuracy

$$\varepsilon(M, \mathcal{P}) = \sup_{(f,X) \in \mathcal{P}} \varepsilon(M(f, X), f, X).$$

The efficiency allows comparing methods in terms of the worst accuracy of their output on any instance of $\mathcal{P}$. However, comparing methods solely on basis of their efficiencies is not fair so far. Indeed, the *computational cost* of two methods may be very different, and comparing efficiencies only make sense for methods with similar costs.

In the context of first-order methods, the main computationally demanding steps performed by the optimization algorithms are usually the evaluations of function values $f(x)$ and gradients $\nabla f(x)$ at different points. Therefore, the computational cost is commonly modelled as proportional to the number of function and gradient evaluations (this is usually approached using the concept of black-box oracle, which is made more precise in Chapter 4 and Chapter 5). Using this model, we are able to rigorously compare methods using their computational costs and their corresponding efficiencies. That is, given a number of

evaluations, a method M_1 performs *better* than a method M_2 if the worst-case inaccuracy of M_1 is smaller than that of M_2, i.e., when $\varepsilon(M_1, \mathcal{P}) \leq \varepsilon(M_2, \mathcal{P})$.

Performing a *global worst-case analysis* of a numerical method consists in characterizing the evolution of its efficiency as a function of its computational cost. Worst-case absolute accuracies are expected to be decreasing functions of the computational cost, and the corresponding decrease rates are usually referred to as the *global convergence rates*.

A standard alternative to the viewpoint taken in the thesis is to consider the computational cost as a function of the required accuracy (see e.g., [NY83]). This is often referred to as *complexity analysis*. For example, given an iterative method M (we denote by $M(.,.;N)$ the method stopped after N iterations) whose computational cost is proportional to the number of iterations N, the aim of global worst-case analyses is to obtain a guarantee on the worst-case absolute accuracy $\varepsilon(M(.,.;N), \mathcal{P})$ (as a function of N). On the other hand, complexity analysis focuses on obtaining guarantees on the computational cost (number of iterations) required to achieve a certain accuracy $\epsilon > 0$:

$$N_{\min}(M, \mathcal{P}; \epsilon) = \min\{N \ : \ \varepsilon(M(f, X; N), f, X) \leq \epsilon \ \forall (f, X) \in \mathcal{P}\}.$$

In the sequel, we use the *worst-case analysis* point of view, but all results can be transposed in terms of complexity analysis.

Classical approaches to convergence analysis are covered in details in numerous seminal references, such as the books of Yudin and Nemirovski [NY83], Polyak [Pol87], Nesterov [Nes04] and the more recent book of Bertsekas [Ber15].

1.3.1 Novel methodologies for global worst-case analyses

As previously underlined, this work is about computing the worst-case absolute inaccuracy of first-order optimization methods. That is, denoting by M some first-order method performing N function and gradient evaluations and by $\mathcal{P}$ the class of optimization problems (f, X) of interest, we want to compute

$$\varepsilon(M, \mathcal{P}) = \sup_{(f,X)\in\mathcal{P}} \varepsilon(M(f, X), f, X) \tag{PEP}$$

The idea of solving (PEP) rose through the work of Drori and Teboulle [DT14]. We believe the main achievements in the development of this novel approach are the following (in historical ordering).

- ◇ Drori and Teboulle [DT14] introduced the idea of performance estimation problems for *smooth unconstrained convex minimization*. The idea is to use semidefinite programming to solve relaxations of the worst-case computation problem (PEP). This allowed to automatically obtain upper

bounds on (PEP) for various optimization schemes in a fully computational way.

- ◇ In the same work, Drori and Teboulle [DT14] managed to devise a new method, by numerically optimizing its worst-case (more precisely, upper bounds on its worst-case).
- ◇ Kim and Fessler [KF16d] managed to find an analytical form for the optimized gradient method (OGM) obtained by Drori and Teboulle. This scheme is shown to have twice better theoretical guarantees compared to standard accelerated methods. However, as only upper bounds were involved so far, it was impossible to conclude whether OGM was optimal or not.
- ◇ In parallel, Lessard, Recht and Packard [LRP16] proposed a cheaper methodology based on control theory for analyzing first-order schemes. This methodology is particularly suited for studying linear convergence rates, and involves semidefinite programs of much smaller dimensions (see discussion in the following section).
- ◇ We (T., Hendrickx and Glineur [THG16a]) proposed a generic way for formulating and solving the performance estimation problems of Drori and Teboulle [DT14] in order to have guaranteed tight results. The new methodology also automatically generates worst-case functions, and not only upper bounds. The main difference with Drori and Teboulle's original approach is essentially threefold: first, we rely on *convex interpolation*, which allows working with finite versions of the convex functions of interest with tightness guarantees. Second, a lifting procedure allows us to work in the primal space (space of functions) and to keep a very transparent approach not relying on multiple relaxations and dualizations. Finally, our approach naturally generalizes to a broader class of problem classes (e.g., involving constraints and non-smooth terms) and methods.

In the next sections, we survey contributions to the performance estimation framework and classify them in three categories. First, we summarize contributions aiming at developing the methodology itself. Then, we survey newly developed algorithms based on the performance estimation framework. Finally, we list other contributions related convergence analyses of optimization methods due to performance estimations.

As this field is very young, this survey is essentially exhaustive. Our main contributions are emphasized throughout the summary.

1.3.2 Survey on the performance estimation framework

Drori and Teboulle [DT14] were first to consider the notion of a performance estimation problem. They focus exclusively on the case of smooth convex

functions equipped with the performance criterion $f(x_N) - f_*$, and introduce the idea of reducing (PEP) to a finite-dimensional problem involving only the iterates x_i, their gradients g_i and function values f_i, along with an optimal point x_* and optimal value f_*. They treat several standard first-order algorithms, namely, the standard fixed-step gradient algorithm, the heavy-ball method [Pol64] and the accelerated gradient method [Nes83]. In their approach, (PEP) is expressed as a non-convex quadratic matrix program [Bec07], which is then relaxed and dualized. The resulting convex problem is then used to provide bounds on the worst-case performance (and, in some cases, is solved analytically). As will be shown later in this work, because of the use of a relaxation and the dualization of a non-convex problem, these bounds are in general not tight, although they turned out to be tight in surprisingly many situations (see Section 4.3). The approach of Drori and Teboulle [DT14] was originally tailored for first-order algorithms minimizing a single smooth convex function over $\mathbb{R}^d$, but an extension to provide upper bounds for the fixed-step projected gradient method is also provided in Drori's thesis [Dro14].

Another computational approach for the analysis and design of first-order algorithms is proposed in [LRP16], in which optimization procedures are regarded as dynamical systems. Integral quadratic constraints (IQC), which are usually used to obtain stability guarantees on complicated dynamical systems, are adapted in order to obtain sufficient conditions for the convergence of optimization algorithms. In a few words, the core idea is to formulate (fixed-step, time-invariant) algorithms as (linear) dynamical systems of the form

$$\begin{aligned} \xi_{k+1} &= A\xi_k + Bu_k, \\ y_k &= C\xi_k + Du_k, \\ u_k &= \nabla f(x_k), \end{aligned}$$

where ξ_k are internal states of the dynamical system at time k, y_k are its outputs, and u_k are its inputs. From this reformulation, the idea is to replace the non-linearity coming from the input ($\nabla f(x_k)$) by (necessary) quadratic constraints (different families of such constraints are developed, see [LRP16, Definition 3]). Then, one can formulate sufficient conditions for the first-order method to converge with rate ρ (in terms of $\|x_k - x_*\|$) as LMI feasibility problems (see [LRP16, Theorem 4]), which can be solved with appropriate solvers and bisection schemes.
This methodology is capable of establishing iteration-independent linear rates of convergence by solving series of small semidefinite programs. However those bounds, valid for any number of iterations, are in general not tight, i.e., more conservative than ours and those of [DT14] when used to estimate worst-case performance after a given finite number of iterations (see Subsection 4.3.1 for an example). In addition, while this methodology is well-suited for studying the linear convergence rates of algorithms for smooth strongly convex optimization, it fails to recover the exact sublinear rates in the non-strongly convex case.

Our contributions (1) — Chapter 4. We (T., Hendrickx, Glineur) formalize the concept of performance estimation problems for smooth unconstrained minimization in [THG16a]. In this work, we use the concept of *convex interpolation* and *smooth (strongly) convex interpolation* in order to transform (PEP) into a semidefinite program without performing any relaxation. That is, any solution to those new problems can be converted to solutions to (PEP). Moreover, the concept of convex interpolation allows us to guarantee that if there exists a convergence proof for a given fixed-step algorithm, then the proof can be obtained using the set of interpolation conditions only.

Our contributions (2) — Chapter 5. Performance estimation problems with tightness guarantees were later extended to larger classes of algorithms, functions and convergence measures in our work [THG16b] (T., Hendrickx, Glineur). In particular, tight guarantees can be obtained for fixed-step algorithm involving projected, proximal and conditional (sub)gradient steps.

1.3.3 Survey on optimized methods

A section of Drori and Teboulle's work [DT14] is also devoted to the optimization of the coefficients of a fixed-step first-order black-box method for smooth unconstrained convex minimization. More precisely, a numerical solver is used to identify a method performing best according to their relaxation of the performance estimation problem, for a known given number of iterations. This approach is taken further in [KF16d], which provides an analytical description of this optimized method. Again we stress that, due to the non-tightness of the relaxation in general, these optimized methods were not guaranteed to have the best possible performances.

Although it is quite remarkable that the optimized gradient method (OGM) has a compact representation similar to the standard fast gradient method (FGM), OGM suffers from a dependence on the number of iterations (more precisely, only the last iterate depends on it). Kim and Fessler [KF16c] therefore studied the convergence of the variants of OGM, which do not depend on the number of iterations, and whose worst-case performances are not significantly worse than that of the original OGM.

Very recently, Drori [Dro16] provided new results related to the optimized gradient method (OGM). In this work, he managed to compute exactly the *minimax risk* (i.e., the best achievable absolute inaccuracy as a function of the number of iterations) of smooth unconstrained convex minimization. This lower bound appears to match the upper complexity bound obtained by Kim and Fessler [KF16d] for OGM, which has therefore the optimal worst-case (for large-scale problems).

After that, Kim and Fessler further studied the worst-case guarantees of optimized gradient-type methods (OGM) and fast gradient methods (FGM) in

terms of residual gradient norm [KF16b] and in their proximal variants [KF16a]. Among others, they show that FGM achieves a $O(N^{-3/2})$ convergence rate for the best gradient norm among iterates (this was premised in our work [THG16a] and verified for a less practical variant of FGM in [Nes12b]), and that it was actually possible to design optimized gradient schemes with better worst-case guarantees than FGM both in function values and residual gradient norms. However, those results do not manage to beat the better $O(N^{-2})$ bound obtained with the regularization technique proposed in [Nes12b] for minimizing the residual gradient norm.

Finally, Drori and Teboulle [DT16] devised an optimal variant of Kelley's cutting plane method for solving non-smooth unconstrained convex minimization problem with Lipschitz objectives. At each iteration of this method, it is possible to chose between two kinds of steps. The first possibility is to solve an intermediate *bundle-like* step, whereas the second one is to perform a simple subgradient step with pre-determined step size (depending on the number of iterations and on the previous bundle-like steps).

Remark 1.1. Note that this research trend that looks towards optimized first-order methods can actually be placed in the broader context of the development of accelerated first-order methods [Nes83, Nes04] and their intuitions. This topic recently attracted much attention. Among others, we refer the reader to the recent works [DFR16, BLS15, AZO14, SBC14, APR15, AP15] aiming at obtaining more intuitions on Nesterov's acceleration both in the smooth strongly convex case and in the degenerate smooth convex case.

1.3.4 Computer-aided convergence analyses

Finally, the performance estimation methodology was used in different works for obtaining new improved (analytical) worst-case guarantees for well-known optimization algorithms; those are summarized in the following paragraphs.

Our contributions (3) — Chapter 6. We (de Klerk, Glineur and T.) analyze in [dKGT16] the steepest descent method with exact line search for minimizing a smooth strongly convex function. In this work, the natural formulation of (PEP) is nonconvex. Therefore, we perform a convex relaxation in order to obtain upper bounds on the worst-case behavior. Those upper bounds turn out to be tight in the case where the line search direction is the gradient and in the case where a relative tolerance on the choice of the direction is allowed.

Our contributions (4) — Chapter 7. We (T., Hendrickx and Glineur) recently applied the performance estimation framework to obtain tight convergence rates for the proximal gradient method in smooth strongly convex

composite minimization [THG16c]. In this work, we emphasize the importance and the differences that may arise by considering different kinds of initial knowledge on the optimization problem under consideration.

Very recently, Shi and Liu [SL16] also used the approach for analyzing cyclic block coordinate descent schemes for smooth unconstrained convex minimization. This work relies on a non-convex formulation and on similar dualization and relaxations as in the original work of Drori and Teboulle [DT14].

1.4 Organization of the thesis

As previously underlined, this thesis is mainly about computing the worst-case guarantees of numerical optimization schemes. The work is organized in three main parts. The first part is dedicated to convex analysis.

- ◇ In Chapter 2, we review the basic definitions, tools and classes of convex functions we use in the thesis. We focus on providing the different elements using a unified approach.
- ◇ Chapter 3 focuses on the development of convex interpolation and integration theorems, which are necessary tools for the subsequent chapters.

The second part is dedicated to the analysis and development of optimization algorithms based on the performance estimation idea and on the tools provided by the convex interpolation results.

- ◇ In Chapter 4, we present the performance estimation framework for the (simpler) smooth unconstrained convex minimization case. This work is published as [THG16a].
- ◇ Chapter 5 focuses on forming an unified framework to treat as much first-order methods as possible in a tight way in the performance estimation framework. This work is contained in [THG16b].
- ◇ In Chapter 6, we provide a tight analysis for the steepest descent algorithm with exact line search (with possibly noise in the search directions) for smooth strongly convex unconstrained minimization. This work is contained in [dKGT16].
- ◇ In Chapter 7, we provide tight convergence analyses of the proximal gradient method for the smooth strongly convex composite minimization. This work is contained in [THG16c].

Part 3 is dedicated to conclusions and further developments.

- ◇ In Chapter 8, we provide new elements and example for further extending the performance estimation framework. The examples include the treatment of monotone inclusions, decentralization, randomness, noise and nonconvexities.
- ◇ Perspectives and conclusions are then drawn in Chapter 9.

1.5 Overview of problem classes and algorithms

The different classes of problems approached throughout the thesis are summarized in Table 1.1. Extensions to decentralized, randomized and second-order algorithms, and to the minimization of finite sums and of nonconvex problems are further discussed in Chapter 8.

Problem type	**Algorithm**
Smooth convex unconstrained minimization	Gradient method (4.3.1) Fast gradient method (4.3.2) Optimized gradient method (4.3.2)
Smooth strongly convex unconstrained minimization	Gradient method (4.3.1, 7.1) Steepest descent (6.1) Inexact direction steepest descent (6.4)
Smooth convex constrained minimization	Conditional gradient method (5.3.5)
Non-smooth convex minimization	Proximal point algorithm (5.3.1) Inexact proximal point algorithm (5.3.1) Projected subgradient method (5.3.2) Proximal gradient method (7.1) Fast proximal gradient method (5.3.3) Proximal optimized gradient method (5.3.4)
Convex set intersection	Alternate projection method (5.3.6) Dykstra alternate projection method (5.3.6)
Monotone inclusions	Forward-backward splitting (8.1) Douglas-Rachford splitting (8.1)

Table 1.1: Summary of methods and problem classes. Most problems are particular instances of the composite convex minimization class (see Section 5.1), and most methods are instances of linear fixed step first-order methods (see Section 5.2.3).

Part I

Discrete Representations of Convex Functions

Chapter 2

Elements of Convex Analysis

This chapter introduces the tools necessary for characterizing discrete representations of convex functions. As it mainly contains standard elements from convex analysis, the reader familiar with the field may safely skip it[1].

We provide basic definitions for the classes of functions and operators of interest for the sequel with a low level of details — we refer the reader to the standard references for convex analysis [HUL96, Roc96, RW98], and to the seminal [BV04, Nes04, Rus06, BL10] (more specifically tailored for optimization) for more detailed treatments and further examples. In this chapter, more details are provided on Legendre-Fenchel duality results, which are used in Chapter 3, and on the different points of view on smoothness and convexity, which are heavily used in the subsequent chapters.

This chapter is organized as follows:

- ◇ Section 2.1 introduces the basic setting we mainly work in for the following chapters: finite dimensional real vector spaces with primal and dual Euclidean structures.
- ◇ In Section 2.2, we provide two standard points of view for approaching convexity of closed sets. Those are termed as *inner* and *outer* views.
- ◇ After that, we treat closed convex functions in Section 2.3 using the two different set convexity points of view.
- ◇ Section 2.4 provides basic intuitions for understanding Legendre-Fenchel conjugation, which plays an important role in Chapter 3.

[1] Except the promotors.

- ◇ Section 2.5 introduces standard classes of functions we work with in the sequel. Those classes are omnipresent in first-order optimization theory.
- ◇ Finally, our main characteristics of interest — smoothness and strong convexity — are strong requirements which are only locally satisfied by most functions. In Section 2.6, we show that it is always possible to extend locally smooth functions by globally smooth ones. With this reasoning, we can use the results related to smooth function with functions that are only locally smooth (which largely widens the class of problems concerned with subsequent analyses).

The main contributions of this chapter are on the one side the presentation of standard results from convex analysis in a unified manner (inner and outer points of view), and the possibility of extending the local strong convexity and smoothness properties of a function to global ones.

2.1 Spaces, norms and scalar products

In the sequel, we work in a finite dimensional real vector space $\mathbb{E}$ and the corresponding dual space $\mathbb{E}^*$ formed by all linear functions on $\mathbb{E}$. The dual pairing is denoted $\langle .,. \rangle : \mathbb{E}^* \times \mathbb{E} \to \mathbb{R}$ (so that for any $s \in \mathbb{E}^*$, the corresponding linear function is denoted $\langle s, . \rangle : \mathbb{E} \to \mathbb{R}$) and satisfies

$$\forall x \in \mathbb{E} \backslash \{0\}, \exists s \in \mathbb{E}^* \text{ such that } \langle s, x \rangle \neq 0,$$
$$\forall s \in \mathbb{E}^* \backslash \{0\}, \exists x \in \mathbb{E} \text{ such that } \langle s, x \rangle \neq 0.$$

We also consider a self-adjoint positive definite linear operator $B : \mathbb{E} \to \mathbb{E}^*$ for $\langle .,. \rangle$, that is, an operator satisfying

$$\langle Bx, y \rangle = \langle By, x \rangle \ \forall x, y \in \mathbb{E} \text{ (self-adjoint)},$$
$$\langle Bx, x \rangle > 0 \ \forall x \in \mathbb{E} \backslash \{0\} \text{ (positive definite)}.$$

This allows us to define the following primal and dual Euclidean norms:

$$\|x\|_{\mathbb{E}}^2 = \langle Bx, x \rangle, \ \forall x \in \mathbb{E}, \quad \|s\|_{\mathbb{E}^*}^2 = \left\langle s, B^{-1}s \right\rangle, \ \forall s \in \mathbb{E}^*;$$

those norms result from the primal and dual scalar products $\langle x, y \rangle_{\mathbb{E}} = \langle Bx, y \rangle$ for $x, y \in \mathbb{E}$ and $\langle x, y \rangle_{\mathbb{E}^*} = \left\langle x, B^{-1}y \right\rangle$ for $x, y \in \mathbb{E}^*$.

Example 2.1. Let $\mathbb{E} = \mathbb{E}^* = \mathbb{R}^d$ and $\langle x, y \rangle$ the standard inner product $\langle x, y \rangle = x^\top y \ \ \forall x, y \in \mathbb{R}^d$. Any positive definite matrix $B \in \mathbb{S}_{++}^d$ induces a pair of primal and dual norms: $\|x\|_{\mathbb{E}}^2 = x^\top Bx$ and $\|x\|_{\mathbb{E}^*}^2 = x^\top B^{-1}x$. The most standard case is to chose B as the identity operator; however, choosing a different scaling matrix B can for example be useful for studying second-order methods

(where the norm is typically defined using the Hessian, see e.g., [Ren01]), or for conditioning purposes (see e.g., [Nes12a] for an application to block coordinate descent-type methods).

Remark 2.2. Note that $\mathbb{E}$ endowed with the inner product $\langle .,. \rangle_{\mathbb{E}}$ (and the corresponding induced norm) is a Hilbert space, and so is $\mathbb{E}^*$ with $\langle .,. \rangle_{\mathbb{E}^*}$. An alternative but equivalent approach to introduce the different inner products is to start with the space $\mathbb{E}$, its reference inner product $\langle .,. \rangle_{\mathbb{E}}$ and the linear operator B, and then define $\langle .,. \rangle$ and $\langle .,. \rangle_{\mathbb{E}^*}$ as by-products.

2.2 Convex sets

The notion of set convexity plays a huge role in optimization theory. The standard intuitive view on convexity of a set is that the set should entirely contain the segment joining any two points of that set. This point of view is referred to as the *inner* characterization of convexity in the sequel.

Definition 2.3. A set $Q \subseteq \mathbb{E}$ is convex if and only if for any $x, y \in Q$ and for any $\lambda \in [0, 1]$:

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

In what follows, we mostly consider non-empty closed convex sets. Those sets have numerous nice properties which renders them very convenient to work with. As an example, the projection operation of any $x \in \mathbb{E}$ onto a non-empty closed convex set Q

$$\Pi_Q(x) = \operatorname*{argmin}_{y \in Q} \|y - x\|_{\mathbb{E}},$$

is always well defined and unique. On the other hand, non-closed sets are often very impractical to work with. Among many possible reasons for that, non-closed sets may turn simple problems into ill-defined ones (i.e., having no solution).

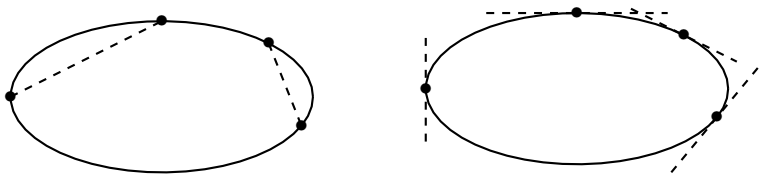
For closed sets, it is possible to use another standard definition of convexity. This alternative to Definition 2.3 uses *outer* point of view, which relies on *supporting closed half-spaces*[2].

Theorem 2.4 (Supporting hyperplanes)**.** Let $Q \subseteq \mathbb{E}$ be a closed set with non-empty interior. Then Q is convex if and only if for every point x_0 of its boundary, there exists an hyperplane $\{x \in \mathbb{E} \mid \langle a, x \rangle = b\}$ (with some $b \in \mathbb{R}$, $a \in \mathbb{E}^*$, $a \neq 0$) such that $\langle a, x \rangle \leq b \ \forall x \in Q$ and $\langle a, x_0 \rangle = b$.

[2]Closed half-spaces are defined as sets of the form $\{x \in \mathbb{E} \mid \langle a, x \rangle \leq b\}$, for some $a \in \mathbb{E}^*$ ($a \neq 0$) and $b \in \mathbb{R}$.

This result is generally referred to as the supporting hyperplane theorem[3], we refer to [BV04, Section 2.5.2] and to [Roc96, Corollary 11.6.1 and Theorem 18.8] for further details.

Example 2.5. Consider the set $Q = \left\{(x, y) \in \mathbb{R}^2 \mid x^2 + 2y^2 \leq 1\right\}$. This set is clearly closed and convex, and therefore has a supporting hyperplane on any of its boundary point, as illustrated in Figure 2.1.



(a) Inner point of view on convexity: every segment joining points of the convex set is in the set.

(b) Outer point of view on convexity: every point on the boundary of the closed convex set has a supporting hyperplane.

Figure 2.1: Illustration of inner and outer points of view on convexity (see Example 2.5).

2.3 Convex functions

As for convex sets, convex functions play a major role in optimization theory. This is essentially because local properties of convex functions (e.g., derivatives) provide global information on the function itself. In general, such global information significantly improves the possibilities for solving the corresponding optimization problems. In this section, we provide two alternative ways (more precisely *conjugate*, or dual ways) for approaching the convexity of closed functions. Those alternatives are essentially the same as for convex sets and rely on the standard definition of convexity (*inner* view) on the one hand (see Definition 2.3), and on supporting hyperplanes (*outer* view) on the other hand (see Theorem 2.4).

Let us start with the basic ingredients for defining convex functions; for any function $f : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$, we associate two sets to it: its domain and its epigraph.

Definition 2.6. Let $f : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$ be a function. The (effective) domain of f, denoted by $\operatorname{dom} f$ is defined as the set:

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

[3]More precisely, this is a converse supporting hyperplane theorem — the set Q is not required to be non-empty for guaranteeing the existence of supporting hyperplanes.

Definition 2.7. Let $f : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$ be a function. The epigraph of f, denoted by $\operatorname{epi} f$ is defined as the set:

$$\operatorname{epi} f = \{(x,t) \in \mathbb{E} \times \mathbb{R} \ : \ t \geq f(x)\}.$$

In order to define convex functions, one first possibility is to rely on the standard idea of using the definition of convexity of a set (see Definition 2.3), which we apply on the epigraph of f.

Definition 2.8. A function $f : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$ is convex if its epigraph is a convex set.

As for convex sets, we mostly restrict ourselves to closed proper functions in the following. Those properties can be defined from the epigraph.

Definition 2.9. A function $f : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$ is proper if its epigraph (or equivalently its domain) is a non-empty set.

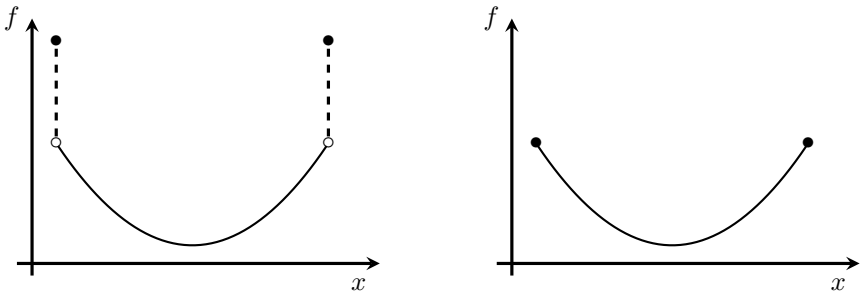
Definition 2.10. A function $f : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$ is closed if its epigraph is a closed set.

Example 2.11. Essentially, non-closed convex functions can only have non-closedness caused by some points on the boundary of their domain (otherwise, the function cannot be convex). Consider for example the following convex function $f : \mathbb{R} \to \mathbb{R}$ (see Figure 2.2)

$$f(x) = \begin{cases} x^2 & \text{if } |x| < 1, \\ 2 & \text{if } |x| = 1, \\ \infty & \text{otherwise.} \end{cases}$$

Clearly, f is a convex function whose epigraph is not a closed set. The function $\tilde{f}$ whose epigraph is the closure of $\operatorname{epi} f$ is given by

$$\tilde{f}(x) = \begin{cases} x^2 & \text{if } |x| \leq 1, \\ \infty & \text{otherwise.} \end{cases}$$



(a) Non-closed convex function f.

(b) Convex function $\tilde{f}$, closure of f.

Figure 2.2: Non-closed convex function f and its closure $\tilde{f}$ (see Example 2.11).

Note that for convex functions, the closedness property is equivalent to the so-called *lower semi-continuity* (e.g., see [Roc96, Theorem 7.1] or [Rus06, Lemma 2.62]).

As for closed convex sets, it is also possible to use an outer approach to define convex functions. This relies on the use of supporting hyperplanes applied to the epigraph (see Theorem 2.4). For convex functions, we usually use non-vertical hyperplanes only, which are in one-to-one correspondence with the so-called *subgradients* (i.e., a non-vertical supporting hyperplane of $\operatorname{epi} f$ is characterized by its normal $(a, a_f) \in \mathbb{E}^* \times \mathbb{R}$ with $a_f \neq 0$ and a constant $b \in \mathbb{R}$, whereas the corresponding subgradient of f is $a \in \mathbb{E}^*$; that is, we discard the dimension corresponding to the values of f in the epigraph). Those subgradients are often used in order to characterize convex functions in the (relative) interior of their domain.

Definition 2.12. An element $g \in \mathbb{E}^*$ is a subgradient of $f : \mathbb{E} \cup \{\infty\} \to \mathbb{R}$ at x if it satisfies $\forall y \in \mathbb{E}$

$$f(y) \geq f(x) + \langle g, y - x \rangle, \tag{2.1}$$

and we denote by $\partial f(x)$ the subdifferential of f at x — i.e., the set of all subgradients of f at x.

Intuitively, subgradients correspond to global linear under-estimators of the function, or more geometrically to non-vertical supporting hyperplanes of the epigraph of f at $(x, f(x))$ — note that f may only have a non-empty subdifferential for points of its domain. Conversely, for any convex function f, the subdifferential $\partial f(x)$ is a non-empty set for any $x \in \operatorname{relint}(\operatorname{dom} f)$ (but may be empty on the boundary, as it may be that the epigraph only has a vertical supporting hyperplane there).

Example 2.13. Let us consider the closed convex function $f : \mathbb{R} \to \mathbb{R}$

$$f(x) = \begin{cases} -\sqrt{-x} & \text{if } x \leq 0, \\ \infty & \text{else.} \end{cases}$$

This function has no subgradient at $x = 0$ (only a vertical supporting hyperplane of epi f is available here). This can be seen on Figure 2.3 and via its derivative $\frac{df}{dx} = \frac{1}{2\sqrt{-x}}$.

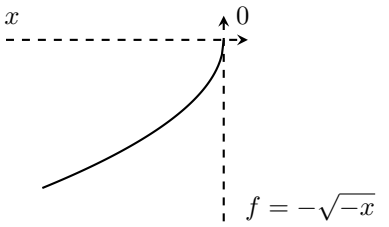


Figure 2.3: Convex function with empty subdifferential at the origin (see Example 2.13).

Note that Definition 2.8 provides a description of a convex function using upper bounds on its values based on neighboring values, whereas it is also possible to define convex functions using lower bounds. This fact is emphasized by the following theorem; as the necessity part of this formulation is not entirely standard, we provide it with a simple proof.

Theorem 2.14. A function $f : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$ is convex if and only if $\forall x \in \text{relint}\,(\text{dom}\, f)$, the set $\partial f(x)$ is non-empty.

Proof. We only prove the necessity part, as it is well-known that if f is convex then $\forall x \in \text{relint}\,(\text{dom}\, f)$, the set $\partial f(x)$ is non-empty[4] — this is essentially the supporting hyperplane theorem (see Theorem 2.4), which imposes that $\forall x \in \text{relint}\,(\text{dom}\, f)$ there exists a non-vertical supporting hyperplane, combined with the fact f is closed on any closed subset of relint $(\text{dom}\, f)$.

(Necessity) Let us prove that $\partial f(x) \neq \emptyset\ \forall x \in \text{relint}\,(\text{dom}\, f)$ implies that f is convex. For any $x_1, x_2 \in \text{dom}\, f$ and $\lambda \in (0, 1)$ we have $y = \lambda x_1 + (1 - \lambda)x_2 \in \text{relint}\,(\text{dom}\, f)$. Therefore, $\exists g \in \partial f(y)$ and

$$f(x_1) \geq f(y) + \langle g, x_1 - y \rangle,$$
$$f(x_2) \geq f(y) + \langle g, x_2 - y \rangle.$$

By combining those inequalities respectively with the coefficients λ and $1 - \lambda$

[4] Among others, one can refer to [Roc96, Theorem 23.4] or [Rus06, Theorem 2.74].

we obtain

$$f(y) \leq \lambda f(x_1) + (1-\lambda) f(x_2),$$

which proves the statement for $\lambda \in (0,1)$. In order to conclude, it is sufficient to notice that the previous inequality trivially holds true for $\lambda = 0$ and $\lambda = 1$. □

The following theorem is central in convex optimization theory; for convex functions, any local optimum is a global optimum.

Theorem 2.15 (Optimality conditions)**.** Let $f : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$ be proper and convex, then $0 \in \partial f(x_*)$ if and only if $x_* = \underset{x \in \mathbb{E}}{\operatorname{argmin}}\ f(x)$.

Proof. Requiring x_* to be optimal is equivalent to require

$$f(x) \geq f(x_*) + \langle 0, x - x_* \rangle \ \forall x \in \mathbb{E}.$$

□

2.4 Legendre-Fenchel conjugation

In this section, we introduce the *Legendre-Fenchel* conjugation, which is essential for the remaining of this work. The common idea behind standard transformations As for common transforms, the use of conjugation is motivated by the fact it renders some operations easier. As examples of such simplifications in the context of other transforms, note that logarithms transform multiplications into additions and Fourier and Laplace transform convolutions into products[5]. The Fenchel-Legendre conjugation on the other hand allows an easier treatment of properties related to differentiability when working in the conjugate space. This transformation dates back to the fifties with the work of Werner Fenchel [Fen49, FB53] generalizing the Legendre transform[6].

In order to introduce the concept, we start with a simple example emphasizing the interpretation of the Legendre transform of a function f as the corresponding function f^* whose gradient ∇f^* is roughly the inverse application of ∇f:

$$y = \nabla f(x) \Leftrightarrow x = \nabla f^*(y).$$

In other words, f and f^* are such that the roles of coordinates and gradients are switched (coordinates and gradients of f respectively becomes gradients

[5]The corresponding *simplified* operations for the Fenchel-Legendre conjugation are the *infimal convolutions*, which are transformed into sums.

[6]More precisely, the Legendre-Fenchel transform is applicable for both convex and non-convex functions, as well as for differentiable and non-differentiable ones. It actually reduces to the Legendre transform in the case of differentiable convex functions (see e.g., [ZRM09] for a recent pedagogical overview with applications in physics).

and coordinates of f^*). We make this statement more precise and introduce the more general Legendre-Fenchel conjugation after the following example.

Example 2.16. Consider the function $f(x) = \frac{a}{2}x^2$, we note $s(x) \overset{\text{(def.)}}{=} \frac{df}{dx}(x) = ax$. By inverting $s(x)$, we obtain $x(s) = \frac{s}{a}$, and $f(x(s)) = \frac{1}{2a}s^2$, which correspond to the value of f given its derivatives.
The Legendre transform of f is a function of s: $f^*(s) \overset{\text{(def.)}}{=} sx(s) - f(x(s)) = \frac{1}{2a}s^2$. This particular definition guarantees that

$$\frac{df^*}{ds}(s) = x(s) + s\frac{dx}{ds}(s) - \frac{df}{dx}(x(s))\frac{dx}{ds}(s).$$

For the case $f(x) = \frac{a}{2}x^2$, this corresponds to $x(s) = \frac{df^*}{ds}(s) = \frac{s}{a}$ when $s = \frac{df}{dx}(x(s)) = ax(s)$. In this case, we indeed conclude that the announced interpretation of the Legendre transform as the transformation linking functions whose derivative are inverse to each others is valid:

$$\left(\frac{df}{dx}(x(s))\right)^{-1} = \frac{df^*}{ds}(s).$$

More precise statements follows from the formal definition of the Legendre-Fenchel conjugation.

Definition 2.17. Let $f : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$ be a function, its Legendre-Fenchel conjugate $f^* : \mathbb{E}^* \to \mathbb{R} \cup \{\infty\}$ is defined as

$$f^*(s) = \sup_{x \in \mathbb{E}} \langle s, x \rangle - f(x). \tag{2.2}$$

Also, we denote by $f^{**} : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$ the biconjugate function

$$f^{**}(x) = \sup_{s \in \mathbb{E}^*} \langle s, x \rangle - f^*(s).$$

Note that conjugate functions are always closed and convex, as they correspond to the maximum of linear functions of s indexed by x. Indeed, the epigraph of f^* is the intersection of the epigraphs of all linear functions of s (indexed by x) $\langle x, s \rangle - f(x)$ — that is, the epigraph of f^* is an intersection of (possibly infinitely many) half-spaces. As the intersection of (either finitely or infinitely many) closed sets is closed, and as the same holds true for the intersection of convex sets, we have that $\operatorname{epi} f^*$ is closed and convex.

Theorem 2.18. Let $f : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$ be a function, its Legendre-Fenchel conjugate $f^* : \mathbb{E}^* \to \mathbb{R} \cup \{\infty\}$ is closed and convex.

Notations 2.19. We denote the set of closed proper convex functions on $\mathbb{E}$ by $\mathcal{F}_{0,\infty}(\mathbb{E})$. The reason for the 0 and ∞ will become clear later, when dealing with smoothness and strong convexity.

Conjugation is often interpreted in terms of affine minorizations. Let $f \in \mathcal{F}_{0,\infty}(\mathbb{E})$ and

$$f^*(z^*) = \sup_{z \in \mathbb{E}} \langle z^*, z \rangle - f(z).$$

For any $z^* \in \mathbb{E}^*$ such that the supremum is finite, we have

$$f(x) \geq \langle z^*, x \rangle - f^*(z^*) \quad \forall x \in \mathbb{E},$$

which provides us with a nice interpretation of $-f^*(z^*)$ as the largest value such that $\langle z^*, . \rangle - f^*(z^*)$ is a global affine minorant of f. In particular,

$$f(x) \geq \langle z^*, x \rangle - f^*(z^*) = f(z) + \langle z^*, x - z \rangle,$$

when the supremum is attained at $z \in \operatorname*{argmax}_{z \in \mathbb{E}} \langle z^*, z \rangle - f(z)$ (i.e., $z^* \in \partial f(z)$). Note that by definition of f^*, we also have

$$f(x) \geq \sup_{z^* \in \mathbb{E}^*} \langle z^*, x \rangle - f^*(z^*),$$

or equivalently $f \geq f^{**}$, with f^{**} having the same affine minorants as f. Therefore, in the case where f is equal to the supremum of all its affine minorants, we have $f = f^{**}$, which is the case for any closed convex function (using the supporting hyperplane theorem, see Theorem 2.4).

In other words, conjugation forms a one-to-one correspondence (an involution) for the class $\mathcal{F}_{0,\infty}$. This is formalized by the following theorem, which is often referred to as the *Fenchel-Moreau* or the *Fenchel biconjugation* theorem (see e.g., [Roc96, Theorem 12.2], [Rus06, Theorem 2.95] or [BL10, Theorem 4.2.1]).

Theorem 2.20. Let $f \in \mathcal{F}_{0,\infty}(\mathbb{E})$. Then, $f^* \in \mathcal{F}_{0,\infty}(\mathbb{E}^*)$ and $f^{**} = f$.

As for the Legendre transform (see Example 2.16), a very standard interpretation of the Legendre-Fenchel conjugation is as an operation reversing the roles of the coordinates and the subgradients: any subgradient (resp. coordinate) of the original function becomes a coordinate (resp. subgradient) of its conjugate.

This interpretation directly results from the application of first-order optimality conditions (Theorem 2.15) to the definition of the conjugation (2.2).

Theorem 2.21. Let $f \in \mathcal{F}_{0,\infty}(\mathbb{E})$ and $f^* \in \mathcal{F}_{0,\infty}(\mathbb{E}^*)$, the following propositions are equivalent conditions on $x \in \mathbb{E}$ and $x^* \in \mathbb{E}^*$:

(a) $f(x) + f^*(x^*) = \langle x^*, x \rangle$,

(b) $x^* \in \partial f(x)$,

(c) $x \in \partial f^*(x^*)$.

Proof. Let $x \in \operatorname*{argmax}_{z \in \mathbb{E}} \langle x^*, z \rangle - f(z)$. By definition of conjugate function (2.2),

we have the following equivalence:

$$x \in \operatorname*{argmax}_{z\in\mathbb{E}} \langle x^*, z\rangle - f(z) \Leftrightarrow f(x) + f^*(x^*) = \langle x^*, x\rangle.$$

In order to prove $(a) \Leftrightarrow (b)$, we can just invoke the necessary and sufficient first-order optimality condition for the optimality of x:

$$x \in \operatorname*{argmax}_{z\in\mathbb{E}} \langle x^*, z\rangle - f(z) \Leftrightarrow 0 \in x^* - \partial f(x) \Leftrightarrow x^* \in \partial f(x).$$

In order to prove the equivalence $(a) \Leftrightarrow (c)$, we use again the definition of conjugate function, which implies

$$f(x) \geq \sup_{s\in\mathbb{E}^*} \langle s, x\rangle - f^*(s).$$

Therefore, Condition (a) is equivalent to $x^* \in \operatorname*{argmax}_{s\in\mathbb{E}^*} \langle s, x\rangle - f^*(s)$, which implies $(a) \Leftrightarrow (c)$ by invoking the necessity and sufficiency of first-order optimality conditions again $x^* \in \operatorname*{argmax}_{s\in\mathbb{E}^*} \langle s, x\rangle - f^*(s) \Leftrightarrow x \in \partial f^*(x^*)$. □

Before going on into the next theoretical facts about conjugation, let us provide several examples.

Example 2.22. (a) Let $a \in \mathbb{R}^d$, the affine function $f(x) = \langle a, x\rangle + b$ is closed and convex, and its conjugate is simply given by

$$f^*(s) = \begin{cases} -b & \text{if } s = a, \\ +\infty & \text{else.} \end{cases}$$

(b) Let $f : \mathbb{R}^d :\to \mathbb{R}$ be the quadratic function $f(x) = \frac{1}{2}\langle Ax, x\rangle$ with $A \in \mathbb{S}^d_{++}$ some positive definite matrix. The function f is closed and convex, and its conjugate is given by $f^*(s) = \frac{1}{2}\langle s, A^{-1}s\rangle$.

(c) Let $\mathbb{E} = \mathbb{E}^* = \mathbb{R}$, with the standard inner product $\langle x, y\rangle = xy$ for $x, y \in \mathbb{R}$. Also consider the ℓ_1-norm $f(x) = |x|$. Its conjugate is

$$f^*(s) = \begin{cases} 0 & \text{if } |s| \leq 1, \\ \infty & \text{else.} \end{cases}$$

Example 2.23. Let us consider the pair of norms $\|.\|_{\mathbb{E}}$ and

$$\|s\|_{\mathbb{E}^*} = \sup_{x\in\mathbb{E}} \{\langle s, x\rangle : \ \|x\|_{\mathbb{E}} \leq 1\},$$

the corresponding conjugate norm and some constant $c > 0$. As we will need it in the sequel, let us prove that $\frac{1}{2c}\|.\|^2_{\mathbb{E}^*}$ can equivalently be seen as the conjugate

of $\frac{c}{2}\|.\|_{\mathbb{E}}^2$. Indeed, we have the following:

$$\sup_{x\in\mathbb{E}} \langle s, x\rangle - \frac{c}{2}\|x\|_{\mathbb{E}}^2 = \max_{\alpha\in\mathbb{R},\|u\|_{\mathbb{E}}\leq 1} \alpha\langle s, u\rangle - \frac{\alpha^2 c}{2},$$

with $\alpha c = \langle s, u\rangle$ by first-order optimality conditions on α. Therefore, we have

$$\begin{aligned}\sup_{x\in\mathbb{E}} \langle s, x\rangle - \frac{c}{2}\|x\|_{\mathbb{E}}^2 &= \frac{1}{2c}\max_{\|u\|_{\mathbb{E}}\leq 1} (\langle s, u\rangle)^2,\\ &= \frac{1}{2c}\left(\max_{\|u\|_{\mathbb{E}}\leq 1} \langle s, u\rangle\right)^2,\\ &= \frac{1}{2c}\|s\|_{\mathbb{E}^*}^2,\end{aligned}$$

where we used the fact that there exists a solution u such that $\langle s, u\rangle \geq 0$ (e.g., $u = 0$) and therefore the maximum is also non-negative, along with the fact that $f(x) = x^2$ is monotonically increasing on $x \geq 0$.

Before going into the next section, let us remark that conjugation also reverses inequalities between functions. This property is crucial as it allows converting lower and upper bounds on closed proper convex functions to respectively upper and lower bounds on their conjugates, and vice versa.

Lemma 2.24. Let $f(x) \geq g(x)\ \forall x \in \mathbb{E}$, then $g^*(s) \geq f^*(s)\ \forall s \in \mathbb{E}^*$.

Proof.

$$f^*(s) = \sup_{x\in\mathbb{E}} \langle s, x\rangle - f(x) \leq \sup_{x\in\mathbb{E}} \langle s, x\rangle - g(x) = g^*(s).$$

□

Using this property, one can note that the definition of convex functions involving upper bounds (Definition 2.8) corresponds to the definition of convex functions using lower bounds (Theorem 2.14) on its conjugate, and reciprocally (more details in the sequel, see Remark 3.5).

2.5 Functional classes

In this section, we introduce the main classes of functions we use in the following chapters; their characteristics were chosen because they are very present nowadays in first-order optimization theory (see e.g., [BTN01, Ber99, Ber09, Ber15, Nes04, Rus06]). The main characteristics of interest for us concern smoothness, strong convexity and gradient and domain boundedness.

2.5.1 Smoothness and strong convexity

The main focus of this section concerns proper closed convex functions satisfying both a smoothness condition and a strong convexity condition. Given two parameters μ and L satisfying $0 \leq \mu \leq L \leq \infty$, we will denote by L the constant characterizing the smoothness (i.e., Lipschitz constant on the gradient), and by μ the strong convexity constant. We will also explicitly allow the case $L = \infty$ to include nonsmooth functions as well, while μ on the other hand is always assumed to be finite. We use the conventions $1/\infty = 0$ and $\infty - \mu = \infty$ to deal with the case $L = \infty$. The main focus of this section (presenting inner and outer points of view for smoothness and strong convexity) is summarized in Table 2.1.

Class	**Definition**	**Inner view**	**Outer view**
Convex set	Definition 2.3	Definition 2.3	Theorem 2.4
Smooth convex	Definition 2.26	Theorem 2.29	Theorem 2.27
Strongly convex	Definition 2.30	Definition 2.30	Theorem 2.32

Table 2.1: Summary of inner and outer points of view on smoothness and strong convexity (for proper, closed and convex functions).

Notations 2.25. Let $L \in \mathbb{R}^+ \cup \{\infty\}$ and $\mu \in \mathbb{R}^+$ be two constants satisfying $\mu \leq L$. We denote by $\mathcal{F}_{\mu,L}(\mathbb{E})$ the set of L-smooth μ-strongly convex closed proper functions $f : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$. As already introduced, we recall the reader that the class of proper closed convex functions is denoted by $\mathcal{F}_{0,\infty}(\mathbb{E})$.

Definition 2.26. Let $f \in \mathcal{F}_{0,\infty}(\mathbb{E})$, and a constant $L \in \mathbb{R}^+ \cup \{\infty\}$. We say that f is L-smooth (notation $f \in \mathcal{F}_{0,L}(\mathbb{E})$) if it satisfies

$$\frac{1}{L}\|g_1 - g_2\|_{\mathbb{E}^*} \leq \|x_1 - x_2\|_{\mathbb{E}} \tag{2.3}$$

for all pairs $x_1, x_2 \in \mathbb{E}$ and corresponding subgradients $g_1, g_2 \in \mathbb{E}^*$ (i.e., such that $g_1 \in \partial f(x_1)$ and $g_2 \in \partial f(x_2)$).

This definition is not entirely standard, as it involves subgradients (even when the function is differentiable) and allows the constant L to be equal to ∞. In the case of a finite L, Condition (2.3) immediately implies uniqueness of the subgradient at each point, hence differentiability of the function, and we recover the well-known Lipschitz condition on the gradient of a smooth function

$$\frac{1}{L}\|\nabla f(x_1) - \nabla f(x_2)\|_{\mathbb{E}^*} \leq \|x_1 - x_2\|_{\mathbb{E}}.$$

On the other hand, when $L = \infty$, the condition becomes vacuous, and the

function can be non-differentiable. The reason for this slightly non-standard definition allowing to choose $L = \infty$ will become clear when dealing with Legendre-Fenchel conjugation of smooth convex functions. An alternative and equivalent way of defining smoothness is to require the function to be upper bounded by its first-order development plus a quadratic term[7].

Theorem 2.27. Let $f \in \mathcal{F}_{0,\infty}(\mathbb{E})$, we have $f \in \mathcal{F}_{0,L}(\mathbb{E})$ if and only if $\forall g \in \mathbb{E}^*$, $\forall x \in \mathbb{E}$ such that $g \in \partial f(x)$, we have $\forall y \in \mathbb{E}$:

$$f(y) \leq f(x) + \langle g, y - x \rangle + \frac{L}{2}\|x - y\|_{\mathbb{E}}^2. \tag{2.4}$$

Note that when a smooth convex function is proper, it has a non-empty relative interior, and hence at least one point satisfying inequality (2.4). Therefore, this function is also defined everywhere (the upper bound (2.4) is finite $\forall x \in \mathbb{E}$), so $\operatorname{dom} f = \mathbb{E}$ and hence $\partial f(x) \neq \emptyset$ $\forall x \in \mathbb{E}$. In addition to that, combining (2.4) with the subgradient inequality (2.1) directly implies uniqueness of g and hence differentiability $g = \nabla f(x)$ when $L < \infty$. This shows that when $L < \infty$, we have (2.4) being strictly equivalent to the well-known inequality usually characterizing smoothness

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|x - y\|_{\mathbb{E}}^2.$$

In addition to that, when $L = \infty$, the condition (2.4) becomes $0 \leq \|x - y\|_{\mathbb{E}}^2$ and hence void (i.e., always satisfied).

As for general convex functions, there are *inner* and *outer* points of view on smoothness (regarding the epigraph, so they respectively correspond to *upper* and *lower* bounds on f) — Theorem 2.27 provides an inner point of view. We start with a simple condition allowing to easily obtain the desired equivalent outer result.

Theorem 2.28. Let $f \in \mathcal{F}_{0,\infty}(\mathbb{E})$, we have $f \in \mathcal{F}_{0,L}(\mathbb{E})$ if and only if $\frac{L}{2}\|x\|_{\mathbb{E}}^2 - f(x) \in \mathcal{F}_{0,\infty}(\mathbb{E})$.

Proof. Let $h(x) = \frac{L}{2}\|x\|_{\mathbb{E}}^2 - f(x)$, $g_f \in \partial f(x)$ and $g_h = LBx - g_f$. First note that f is closed and proper if and only if h is closed and proper. Then, the result follows from the verification of the equivalence

$$f(y) \leq f(x) + \langle g_f, y - x \rangle + \frac{L}{2}\|x - y\|_{\mathbb{E}}^2 \Leftrightarrow h(y) \geq h(x) + \langle g_h, y - x \rangle \quad \forall x, y \in \mathbb{E},$$

[7] We do not provide it with a proof, as it follows from the same reasoning as in standard references — that is, re-write the function with its zeroth-order development plus an integral involving the first order term, and then bound the first-order term using (2.26) — see e.g., [Nes04, Lemma 1.2.3, Theorem 2.1.5].

with $g_h \in \partial h(x)$ and $\partial h(x) \neq \emptyset \ \forall x \in \mathbb{E}(= \operatorname{dom} h)$ if and only if $h \in \mathcal{F}_{0,\infty}$ (see Theorem 2.14). □

The following theorem provides an outer description of smooth functions; it is illustrated in Figure 2.4(a),

Theorem 2.29. Let $f \in \mathcal{F}_{0,\infty}(\mathbb{E})$. We have that $f \in \mathcal{F}_{0,L}(\mathbb{E})$ if and only if $\forall x, y \in \mathbb{E}$:

$$f(\lambda x + (1-\lambda)y) \geq \lambda f(x) + (1-\lambda)f(y) - \lambda(1-\lambda)\frac{L}{2}\|x-y\|_{\mathbb{E}}^2. \tag{2.5}$$

Proof. Let $f \in \mathcal{F}_{0,\infty}(\mathbb{E})$ and denote $h(x) = \frac{L}{2}\|x\|_{\mathbb{E}}^2 - f(x)$ (with $h \in \mathcal{F}_{0,\infty}$ if and only if $f \in \mathcal{F}_{0,L}$ by Theorem 2.28). The results follows from the following equivalence:

$$\begin{aligned} &h(\lambda x + (1-\lambda)y) \leq \lambda h(x) + (1-\lambda)h(y) \\ &\quad \Leftrightarrow f(\lambda x + (1-\lambda)y) \geq \lambda f(x) + (1-\lambda)f(y) - \lambda(1-\lambda)\frac{L}{2}\|x-y\|_{\mathbb{E}}^2, \end{aligned}$$

$\forall x, y \in \mathbb{E}$ and $\forall \lambda \in [0,1]$. □

Strong convexity, on the other hand, is a strengthening of the convexity condition, that can also be seen both in terms of outer and inner bounds defining convex functions. Let us start with the inner bound.

Definition 2.30. Let $f \in \mathcal{F}_{0,\infty}(\mathbb{E})$ and a constant $\mu \in \mathbb{R}^+$. We say that f is μ-strongly convex (notation $f \in \mathcal{F}_{\mu,\infty}(\mathbb{E})$) if it satisfies for any $x, y \in \mathbb{E}$ and for any $\lambda \in [0,1]$:

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y) - \lambda(1-\lambda)\frac{\mu}{2}\|x-y\|_{\mathbb{E}}^2. \tag{2.6}$$

The following theorem proposes an alternative characterization for strongly convex functions, and can be deduced from Definition 2.6 along with the Euclidean structure of the primal norm $\|.\|_{\mathbb{E}}$.

Theorem 2.31. $f \in \mathcal{F}_{\mu,\infty}(\mathbb{E})$ if and only if $f(x) - \frac{\mu}{2}\|x\|_{\mathbb{E}}^2 \in \mathcal{F}_{0,\infty}(\mathbb{E})$.

This alternative definition can be used to obtain a third way of seeing strongly convex functions, using an outer characterization, similarly to what is proposed by Theorem 2.14 for convex functions.

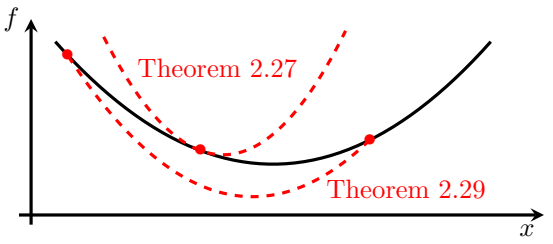
Theorem 2.32. Let $f \in \mathcal{F}_{0,\infty}(\mathbb{E})$, then $f \in \mathcal{F}_{\mu,\infty}(\mathbb{E})$ if and only if

$$f(y) \geq f(x) + \langle g, y-x \rangle + \frac{\mu}{2}\|x-y\|_{\mathbb{E}}^2, \tag{2.7}$$

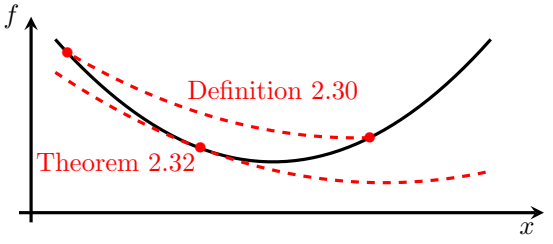
$\forall y \in \mathbb{E}$ and $\forall x \in \mathbb{E}$, $\forall g \in \mathbb{E}^*$ such that $g \in \partial f(x)$.

Using the previous definitions, one can readily extend Theorem 2.31 for handling smoothness, in addition to strong convexity. This can easily be obtained again by using the Euclidean structure of $\|.\|_{\mathbb{E}}$, along with the upper bound 2.4 coming from the smoothness assumption.

Theorem 2.33. Consider a function $f \in \mathcal{F}_{0,\infty}(\mathbb{E})$. We have that $f \in \mathcal{F}_{\mu,L}(\mathbb{E})$ if and only if $f(x) - \frac{\mu}{2}\|x\|_{\mathbb{E}}^2 \in \mathcal{F}_{0,L-\mu}(\mathbb{E})$.



(a) Inner and outer point of view on smoothness for closed convex functions.



(b) Inner and outer point of view on strong convexity for closed convex functions.

Figure 2.4: Inner and outer views on strong convexity and smoothness.

Finally, smoothness and strong convexity are closely tied by the Legendre-Fenchel duality — this is a key element for the developments presented in the following chapters. Although very standard[8], we provide a more compact proof of this duality result here, which totally fits our original smoothness and strong convexity definitions — although very similar, the results presented in [KSST09, Theorem 6] do not share exactly the same setting.

Theorem 2.34. Consider a function $f \in \mathcal{F}_{0,\infty}(\mathbb{E})$. We have $f \in \mathcal{F}_{0,L}(\mathbb{E})$ if and only if $f^* \in \mathcal{F}_{1/L,\infty}(\mathbb{E}^*)$.

Note that in the case $L = \infty$, this theorem reduces to Theorem 2.20. This duality result explains why we need to include the case $L = \infty$ in our original

[8] We refer to [RW98, Proposition 12.60] for the standard setting with self-conjugate norms $\|.\|_{\mathbb{E}} = \|.\|_{\mathbb{E}^*}$ and to [KSST09, Theorem 6] for the dual norm setting — using slightly different definitions.

definition of smoothness: this is so that we can include the conjugates of smooth but non-strongly convex functions in $\mathcal{F}_{0,L}(\mathbb{E})$.

Proof. The idea of the proof is to show the equivalence between having a quadratic upper bound on f and having a quadratic lower bound on f^*. The key ingredient we use in the proof is the result from Lemma 2.24 stating that lower and upper bounds are reversed when using conjugation. That is, $\forall f, g : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$ such that $f(x) \geq g(x)\ \forall x \in \mathbb{E}$, we have $g^*(y) \geq f^*(y)$ $\forall y \in \mathbb{E}^*$.

The proof is divided in three steps.

(a) We start by defining the function $h(\Delta x) = f(x+\Delta x) - (f(x) + \langle g, \Delta x\rangle)$, for any x and any g such that $g \in \partial f(x)$. Note that a L-smoothness assumption on f corresponds to $h(\Delta x) \leq \frac{L}{2}\|\Delta x\|^2_{\mathbb{E}}$, $\forall \Delta x \in \mathbb{E}$ (see Theorem 2.27).

(b) Using together Lemma 2.24 along with norm conjugation (see Example 2.23), one can note that this upper-bound requirement on h is equivalent to the lower-bound requirement $h^*(\Delta g) \geq \frac{1}{2L}\|\Delta g\|^2_{\mathbb{E}^*}$, $\forall \Delta g \in \mathbb{E}^*$.

(c) We show that the condition $h^*(\Delta g) \geq \frac{1}{2L}\|\Delta g\|^2_{\mathbb{E}^*}$ is equivalent to the $1/L$-strong convexity of f^*.

From the reasoning, it remains to prove (c); we focus on proving the equivalence between $h^*(\Delta g) \geq \frac{1}{2L}\|\Delta g\|^2_{\mathbb{E}^*}$ and $1/L$-strong convexity of f^* using Theorem 2.32. That is, we have to show that[9]

$$f^*(g + \Delta g) \geq f^*(g) + \langle \Delta g, x\rangle + \frac{1}{2L}\|\Delta g\|^2_{\mathbb{E}^*},$$

$\forall \Delta g \in \mathbb{E}^*$ and $\forall g \in \mathbb{E}^*$, $\forall x \in \mathbb{E}$ such that $x \in \partial f^*(g)$ (or equivalently $\forall x \in \mathbb{E}$,$\forall g \in \mathbb{E}^*$ such that $g \in \partial f(x)$ by Theorem 2.21). To this aim, we only have to obtain an expression of h^* in terms of f^*. The following lines are valid

[9] We only show the strong convexity requirement, as we already know that f^* is closed and convex (Theorem 2.20) and therefore already satisfies the condition $\partial f^*(g) \neq \emptyset\ \forall g \in$ relint (dom f^*) (Theorem 2.14).

$\forall x \in \mathbb{E}, \forall g \in \mathbb{E}^*$ such that $g \in \partial f(x)$:

$$\begin{aligned}
h^*(\Delta g) &= \sup_{\Delta x \in \mathbb{E}} \langle \Delta g, \Delta x \rangle - h(\Delta x), \\
&= f(x) + \sup_{\Delta x \in \mathbb{E}} \langle g + \Delta g, \Delta x \rangle - f(x + \Delta x), \text{ (definition of } h) \\
&= f(x) + \sup_{x' \in \mathbb{E}} \langle g + \Delta g, x' - x \rangle - f(x'), \text{ (new variable } x' = x + \Delta x) \\
&= f(x) - \langle g + \Delta g, x \rangle + \sup_{x' \in \mathbb{E}} \langle g + \Delta g, x' \rangle - f(x'), \\
&= f(x) - \langle g + \Delta g, x \rangle + f^*(g + \Delta g), \text{ (definition of } f^*) \\
&= f^*(g + \Delta g) - f^*(g) - \langle \Delta g, x \rangle \\
&\quad \text{(using the assumption } g \in \partial f(x), \text{ equivalent to } f(x) - \langle x, g \rangle = -f^*(g) \\
&\quad \text{by Theorem 2.21).}
\end{aligned}$$

Therefore, we have the claim

$$\frac{1}{2L} \|\Delta g\|_{\mathbb{E}^*}^2 \leq f^*(g + \Delta g) - f^*(g) - \langle \Delta g, x \rangle,$$

$\forall \Delta g \in \mathbb{E}^*$ and $\forall g \in \mathbb{E}^*$,$\forall x \in \mathbb{E}$ such that $x \in \partial f^*(g)$.

□

Corollary 2.35. Consider a function $f \in \mathcal{F}_{0,\infty}(\mathbb{E})$. We have $f \in \mathcal{F}_{\mu,L}(\mathbb{E})$ if and only if $f^* \in \mathcal{F}_{1/L,1/\mu}(\mathbb{E}^*)$.

Example 2.36. Let us consider the case $\mathbb{E} = \mathbb{E}^* = \mathbb{R}^d$, with the standard inner product $\langle x, y \rangle = x^\top y$ for $x, y \in \mathbb{R}^d$. Also consider the positive definite matrix $A \succ 0$, the corresponding quadratic function $f(x) = \frac{1}{2} x^\top A x$ and its conjugate $f^*(z) = \frac{1}{2} z^\top A^{-1} z$ (see Example 2.22).

By denoting $\mu_f = \lambda_{\min}(A)$ and $L_f = \lambda_{\max}(A)$ respectively the minimum and maximum eigenvalues of A, one can note that f is μ_f-strongly convex and L_f-smooth.

Similarly, by denoting $\mu_{f^*} = \lambda_{\min}(A^{-1})$ and $L_{f^*} = \lambda_{\max}(A^{-1})$, we have that f^* is μ_{f^*}-strongly convex and L_{f^*}-smooth, with $\mu_{f^*} = L_f^{-1}$ and $L_{f^*} = \mu_f^{-1}$.

Example 2.37. Fenchel duality between strong convexity and smoothness is a fundamental principle used for example for smoothing techniques arising in optimization (i.e., approximating a non-smooth convex objective function by a smooth one); see for example [Nes05, DGN12, BT12]. Essentially, the idea is to regularize the conjugate f^* of the initially non-smooth function f with a strongly convex function and then to approximate the original function by the conjugate of the regularized f^*. For instance, regularizing f^* by choosing $\tilde{f}^*(g) = f^*(g) + \frac{\tau}{2} \|g\|_{\mathbb{E}^*}^2$ and then approximate f by $\tilde{f} = \tilde{f}^{**}$.

As an example, let $\mathbb{E} = \mathbb{E}^* = \mathbb{R}$, with the standard inner product $\langle x, y \rangle = xy$ for $x, y \in \mathbb{R}$. Also consider the ℓ_1-norm $f(x) = |x|$ as the non-smooth function.

Its conjugate is

$$f^*(z) = \begin{cases} 0 & \text{if } |z| \leq 1, \\ \infty & \text{else.} \end{cases}$$

We denote by $h^*(z) = f^*(z) + \frac{\mu}{2}|z|^2$ a regularization of f^* with $\mu \geq 0$. The corresponding conjugate is:

$$h(x) = \begin{cases} \frac{x^2}{2\mu} & \text{if } |x| < \mu, \\ |x| - \frac{\mu}{2} & \text{else,} \end{cases}$$

which is $1/\mu$-smooth (see Figure 2.37). This kind of functions is usually referred to as a Huber loss (see e.g. [Hub64]); it is commonly used in regressions for approximating the ℓ_1-regularization.

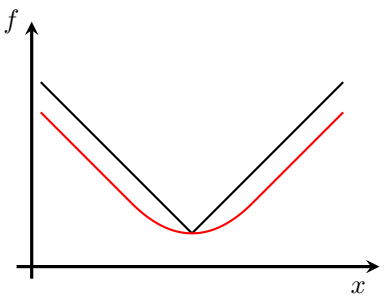


Figure 2.5: Absolute value (black) and its smoothing (red).

Finally, the following theorem provide a simple criterion to check whether a twice differentiable function is μ-strongly convex and L-smooth. For that purpose, we respectively consider the lowest and largest eigenvalues of a linear application $A : \mathbb{E} \to \mathbb{E}^*$: respectively $\lambda_{\min}(A) = \min_{x \in \mathbb{E}} \frac{\langle Ax, x \rangle}{\|x\|_{\mathbb{E}}^2}$ and $\lambda_{\max}(A) = \max_{x \in \mathbb{E}} \frac{\langle Ax, x \rangle}{\|x\|_{\mathbb{E}}^2}$. The proof is very standard, and we do not provide it. The interested reader can refer to e.g., [Nes04, Lemma 1.2.2] for the smooth part, and use the same idea for the strongly convex one.

Theorem 2.38. Let $f : \mathbb{E} \to \mathbb{R}$ being twice continuously differentiable. Then $f \in \mathcal{F}_{\mu,L}(\mathbb{E})$ if and only if $\mu \leq \lambda_{\min}(\nabla^2 f(x))$ and $\lambda_{\max}(\nabla^2 f(x)) \leq L \ \forall x \in \mathbb{E}$.

2.5.2 Domain and subgradient boundedness

Definition 2.39. Let $f \in \mathcal{F}_{0,\infty}(\mathbb{E})$ and a constant $M \in \mathbb{R}^+ \cup \{\infty\}$. We say that f has M-bounded subgradients (resp. subgradient variations) if it satisfies

$$\|g\|_{\mathbb{E}^*} \leq M \text{ (resp. } \|g_1 - g_2\|_{\mathbb{E}^*} \leq M),$$

$\forall g \in \cup_{x\in\mathbb{E}} \partial f(x) (= \operatorname{dom} f^*)$ (resp. $\forall g_1, g_2 \in \operatorname{dom} f^*$).

Closed convex functions with bounded subgradient are often referred to as Lipschitz functions, as they are often defined using the relation $\forall x, y \in \operatorname{dom} f$:

$$|f(x) - f(y)| \leq M\|x - y\|_{\mathbb{E}}.$$

(this can simply be obtained from the subgradient inequality and Definition 2.39). The class of closed and proper convex functions with bounded subgradient variations naturally appears as the class of function conjugate to the class of function with bounded domain, whereas having bounded gradients instead corresponds to the class conjugate to functions with bounded domain centered at the origin (more details in the sequel).

Remark 2.40. Convex functions with bounded subgradient variations are within the family of convex functions with bounded subgradients (with different constants), and reciprocally — this can be obtained by an appropriate use of triangle inequalities along with Definition 2.39).

Definition 2.41. Let $f \in \mathcal{F}_{0,\infty}(\mathbb{E})$ and a constant $M \in \mathbb{R}^+ \cup \{\infty\}$. We say that f has a radius M (resp. a diameter M) if it satisfies

$$\|x\|_{\mathbb{E}} \leq M \text{ (resp. } \|x_1 - x_2\|_{\mathbb{E}} \leq M),$$

for any $x \in \operatorname{dom} f$ (resp. $x_1, x_2 \in \operatorname{dom} f$).

Note that as in the case of smoothness, the boundedness constant M is also allowed to take the value ∞, in order to embed the unbounded (domain or gradient) cases.

Notations 2.42. We denote by $\mathcal{C}_{M,L}(\mathbb{E})$ (resp. $\mathcal{C}'_{M,L}(\mathbb{E})$) the class of closed proper convex L-smooth functions with M-bounded subgradients (resp. subgradient variations) and by $\mathcal{S}_{M,\mu}(\mathbb{E})$ (resp. $\mathcal{S}'_{M,\mu}(\mathbb{E})$) the class of closed proper μ-strongly convex functions with a radius (resp. diameter) M.

As for smoothness and strong convexity, domain and gradient boundedness are closely related via Legendre-Fenchel conjugation. This is very natural in view of the interpretation of conjugation as an operation reversing the roles of coordinates and subgradients (see Theorem 2.21), and is in particular emphasized by the next theorem.

Theorem 2.43. Let $f \in \mathcal{F}_{0,\infty}(\mathbb{E})$. We have $f \in \mathcal{S}_{M,0}(\mathbb{E})$ (resp. $f \in \mathcal{S}'_{M,0}(\mathbb{E})$) if and only if $f^* \in \mathcal{C}_{M,V}(\mathbb{E}^*)$ (resp. $f^* \in \mathcal{C}'_{M,\infty}(\mathbb{E}^*)$).

Proof. We focus on the case $M < \infty$, as the result trivially holds otherwise. Also, we only detail the case of $f \in \mathcal{S}_{M,0}(\mathbb{E})$ and $f^* \in \mathcal{C}_{M,\infty}(\mathbb{E}^*)$ as the corresponding results with bounded variations and domains follow from a similar reasoning.

Let us start with the case $f \in \mathcal{S}_{M,0}(\mathbb{E})$. In that setting we have that $\forall x \in \mathbb{E}, \forall g \in \mathbb{E}^*$ such that $g \in \partial f(x)$, x is such that $\|x\|_{\mathbb{E}} \leq M$ (because $\partial f(x) \neq \emptyset$ only for $x \in \operatorname{dom} f$). The previous statement is equivalent to $\forall x \in \mathbb{E}, \forall g \in \mathbb{E}^*$ such that $x \in \partial f^*(g)$ we have $\|x\|_{\mathbb{E}} \leq M$. Hence, $f \in \mathcal{S}_{M,0}(\mathbb{E}) \Rightarrow f^* \in \mathcal{C}_{M,\infty}(\mathbb{E}^*)$.

Second, let $f^* \in \mathcal{C}_{M,\infty}(\mathbb{E}^*)$. In that setting we have that $\forall x \in \mathbb{E}, \forall g \in \mathbb{E}^*$ such that $x \in \partial f^*(g)$, x is such that $\|x\|_{\mathbb{E}^*} \leq M$. This is equivalent to $\forall x \in \mathbb{E}, \forall g \in \mathbb{E}^*$ such that $g \in \partial f(x)$, $\|x\|_{\mathbb{E}} \leq M$. This proves that $\forall x \in \operatorname{relint}(\operatorname{dom} f)$ we have $\|x\|_{\mathbb{E}} \leq M$ (see Theorem 2.14).

In order to prove that $\|x\|_{\mathbb{E}} \leq M$ $\forall x \in \operatorname{dom} f$, we consider two cases: (a) $\operatorname{relint}(\operatorname{dom} f) \neq \emptyset$ and (b) $\operatorname{relint}(\operatorname{dom} f) = \emptyset$.

(a) We proceed by contradiction. In this case, let us consider some $x \in \mathbb{E}$: $\|x\| > M$ and let some $y \in \operatorname{relint}(\operatorname{dom} f)$. Then, there exists $\lambda \in (0,1)$ such that $\|z\|_{\mathbb{E}} > M$ with $z = \lambda x + (1-\lambda)y$ (so $z \in \operatorname{relint}(\operatorname{dom} f)$). Therefore, there exists $g \in \partial f(z)$ (see Theorem 2.14), and by conjugation $z \in \partial f^*(g)$ is such that $\|z\| > M$ which is a contradiction with the assumption $f^* \in \mathcal{C}_{M,\infty}(\mathbb{E}^*)$.

(b) Let us consider an empty relative interior. That is, $\operatorname{dom} f$ is a singleton. Then, for $x \in \operatorname{dom} f$ and $\forall g \in \mathbb{E}$ we have $g \in \partial f(x)$, and therefore $x \in \partial f^*(g)$ and hence $\|x\|_{\mathbb{E}} \leq M$ by assumption.

Hence when $f \in \mathcal{F}_{0,\infty}(\mathbb{E})$ we have $f^* \in \mathcal{C}_{M,\infty}(\mathbb{E}^*) \Rightarrow f \in \mathcal{S}_{M,0}(\mathbb{E})$, which concludes the proof.

□

Corollary 2.44. Consider a function $f \in \mathcal{F}_{0,\infty}(\mathbb{E})$. We have $f \in \mathcal{C}_{M,L}(\mathbb{E})$ (resp. $f \in \mathcal{C}'_{M,L}(\mathbb{E})$) if and only if $f^* \in \mathcal{S}_{M,1/L}(\mathbb{E}^*)$ (resp. $f^* \in \mathcal{S}'_{M,1/L}(\mathbb{E}^*)$).

Remark 2.45. Note that not all those properties — that is, smoothness, strong convexity and gradient and domain boundedness — are compatible with each other. For example, smoothness and bounded gradient imposes that $\operatorname{dom} f = \mathbb{E}$, and can therefore clearly not be combined with domain boundedness (with $M < \infty$). On the other hand, the strong convexity requirement is not compatible with bounded subgradients, as for any strongly convex function $f \in \mathcal{F}_{\mu,\infty}$ (with $\mu > 0$) and for any vector $g \in \mathbb{E}^*$, there exists $x \in \mathbb{E}$ such that $g \in \partial f(x)$ — which cannot be combined with the fact that any subgradient should have a bounded norm. Another way to understand the incompatibility

between strong convexity and bounded gradients is to remark that its conjugate would be a smooth convex function with bounded domain, which is impossible.

Those incompatibilities originate from the fact we require the functions to *globally satisfy* boundedness, smoothness, and strong convexity properties, instead of only locally (local versions of smoothness and strong convexity are handled in Section 2.6).

As examples of smooth functions with bounded subgradients and strongly convex functions with bounded domain, one can consider the Huber function $h(x)$ of Example 2.37, and its conjugate $h^*(x)$.

2.5.3 Indicator and support functions

Constraints are so recurrent in optimization that we dedicate the next lines specifically to them. For that, we first introduce the class of indicator function, and then the class of support functions, the class of convex conjugates to indicator functions.

Definition 2.46. Let $Q \subseteq \mathbb{E}$ be a closed convex set and define $i_Q : \mathbb{E} \to \{0, \infty\}$:

$$i_Q(x) = \begin{cases} 0 & \text{if } x \in Q, \\ \infty & \text{otherwise.} \end{cases}$$

We call $i_Q : \mathbb{E} \to \{0, \infty\}$ the indicator function of Q.

Notations 2.47. We denote by $\mathcal{I}_M(\mathbb{E})$ the class of closed convex indicator functions that are bounded in terms of radius, and alternatively $\mathcal{I}'_M(\mathbb{E})$ for those bounded in terms of diameter.

Definition 2.48. Let $Q \subseteq \mathbb{E}$ be a closed convex set and define $\sigma_Q : \mathbb{E}^* \to \mathbb{R} \cup \{\infty\}$

$$\sigma_Q(s) = \sup_{x \in Q} \langle s, x \rangle.$$

We call $\sigma_Q : \mathbb{E}^* \to \mathbb{R} \cup \{\infty\}$ the support function of Q.

Intuitively, support functions provide correspondences between a supporting hyperplane $s \in \mathbb{E}^*$ and the distance (in the direction of s) to the origin of the point on which it supports the convex set Q.

Note that support functions can be seen as naturally defined as convex conjugate to indicator functions, as we can write:

$$\sigma_Q(s) = \sup_{x \in Q} \langle s, x \rangle = \sup_{x \in \mathbb{E}} \langle s, x \rangle - i_Q(x).$$

As a consequence, M-bounded indicator function are convex conjugate to support functions with M-bounded subgradients (by Theorem 2.43).

Notations 2.49. The set of support functions with bounded subgradients is denoted by $\mathcal{I}^*_M(\mathbb{E}^*)$ and $\mathcal{I}'^*_M(\mathbb{E}^*)$ for the support functions that have bounded subgradient variations.

Example 2.50. Every norm $\|.\|_{\mathbb{E}^*}$ is the support function of the unit ball of its conjugate norm $\|.\|_{\mathbb{E}}$ by definition of conjugate norm: $\|z\|_{\mathbb{E}^*} = \sup_{\|x\|_{\mathbb{E}}\leq 1} \langle z, x\rangle$.

2.5.4 Non-convex smooth functions

Let us now leave the convex world for a moment. We introduce a class of differentiable non-convex functions. This class of function has the particularity of being definable using convex functions, and is also very present in optimization theory and practice.

Definition 2.51. A differentiable function $f : \mathbb{E} \to \mathbb{R}$ is L-smooth if and only if it satisfies the following condition $\forall x, y \in \mathbb{E}$:

$$|f(x) + \langle \nabla f(x), y - x\rangle - f(y)| \leq \frac{L}{2}\|x - y\|^2_{\mathbb{E}}. \tag{2.8}$$

Also note that it is alternatively possible to define this class of function using a standard Lipschitz condition on the gradient $\forall x, y \in \mathbb{E}$ (when $L < \infty$):

$$\frac{1}{L}\|\nabla f(x) - \nabla f(y)\|_{\mathbb{E}^*} \leq \|x - y\|_{\mathbb{E}}.$$

Notations 2.52. In order to denote smooth non-linear functions, we overload the notation used for closed proper smooth strongly convex functions. That is, the class of L-smooth non-linear functions over $\mathbb{E}$ is denoted $f \in \mathcal{F}_{-L,L}(\mathbb{E})$.

The following lemma allows obtaining simple alternative definition for this class of function using the class of smooth convex functions.

Lemma 2.53. Let $f : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$. We have $f \in \mathcal{F}_{-L,L}(\mathbb{E}) \Leftrightarrow f + \frac{L}{2}\|x\|^2_{\mathbb{E}} \in \mathcal{F}_{0,2L}(\mathbb{E})$.

Proof. Let $f : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$ and define $h(x) = f(x) + \frac{L}{2}\|x\|^2_{\mathbb{E}}$. We have that $\nabla h(x) = \nabla f(x) + LBx$, and $\forall x, y \in \mathbb{E}$:

$$\begin{aligned} f(x) + \langle \nabla f(x), y - x\rangle - f(y) \leq \frac{L}{2}\|x - y\|^2_{\mathbb{E}} &\Leftrightarrow h(y) \geq h(x) + \langle \nabla h(x), y - x\rangle, \\ -f(x) - \langle \nabla f(x), y - x\rangle + f(y) \leq \frac{L}{2}\|x - y\|^2_{\mathbb{E}} &\Leftrightarrow h(y) \leq h(x) + \langle \nabla h(x), y - x\rangle \\ &\quad + L\|x - y\|^2_{\mathbb{E}}, \end{aligned}$$

where the equivalences are obtained by expressing f and ∇f in terms of h and ∇h (or reciprocally), which proves our statement. □

Remark 2.54. Note that smooth convex functions from Definition 2.51 could alternatively be defined using different constants for characterizing upper and lower quadratic bounds.

The following theorem proves an useful characterization of twice differenciable L-smooth functions.

Theorem 2.55. Let $f : \mathbb{E} \to \mathbb{R}$ be twice continuously differentiable. Then $f \in \mathcal{F}_{-L,L}(\mathbb{E})$ if and only if $-LI \leq \lambda_{\min}(\nabla^2 f(x))$ and $\lambda_{\max}(\nabla^2 f(x)) \leq L$ $\forall x \in \mathbb{E}$.

The proof of this theorem is very standard and we therefore not provide it here (as for Theorem 2.38, the reader can refer to e.g., [Nes04, Lemma 1.2.2]).

Example 2.56. Consider the case $\mathbb{E} = \mathbb{E}^* = \mathbb{R}^d$ with the standard Euclidean inner product. Let f be a quadratic function $f(x) = \frac{1}{2}x^\top Ax$ defined from the symmetric matrix $A \in \mathbb{S}^d$ with bounded eigenvalues $-LI_d \preceq A \preceq LI_d$. We have that $f \in \mathcal{F}_{-L,L}(\mathbb{R}^d)$.

2.6 Local and global smoothness

For practical optimization problems, smoothness and strong convexity can be seen as very strong requirements. However, we can in general be satisfied even when those requirements are only locally met on a subset of the domain. One common example of that situation is the case of constrained optimization problems for which the objective function satisfies some smoothness condition on its domain (but possibly not outside of it). This situation is very common, as for example any differentiable function defined on a compact set locally satisfies a smoothness condition.

In order to show that this local property suffices, we provide a construction for generating a smooth extension to the function. We start by assuming that the function f satisfies the following smoothness condition (2.4) on some closed convex domain Q:

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|x - y\|_{\mathbb{E}}^2 : \quad \forall x, y \in Q.$$

We show that it is also valid for the other definitions of smoothness and discuss similar ideas for obtaining global extensions for strong convexity and non-convex smooth functions afterwards (see Corollary 2.60 and Corollary 2.61 in the sequel).

Theorem 2.57. Let $f : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$ be closed and proper, and $Q \subseteq \mathbb{E}$ be a closed convex set. Also, assume f being finite on Q and such that for all $x \in Q$, there exists $g_x \in \mathbb{E}^*$ satisfying

$$f(y) \geq f(x) + \langle g_x, y - x \rangle \qquad : \forall y \in Q, \tag{2.9}$$

$$f(y) \leq f(x) + \langle g_x, y - x \rangle + \frac{L}{2}\|x - y\|_{\mathbb{E}}^2 \qquad : \forall y \in Q. \tag{2.10}$$

Then, there exists a function $\tilde{f} \in \mathcal{F}_{0,L}(\mathbb{E})$ such that $\tilde{f}(x) = f(x)$ and $\nabla \tilde{f}(x) = g_x$ for all $x \in Q$.

Proof. First, note that for any $x \in Q$, g_x is necessarily unique; so we denote $g_x = \nabla f(x)$ in the following.

Let $y \in Q$ and $x \in \mathbb{E}$, we define the local upper bound on f obtained from y:

$$q_y(x) = f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2}\|x - y\|_{\mathbb{E}}^2.$$

Also, we define the function $\tilde{f}(x)$:

$$\tilde{f}(x) = \operatorname{convh}\{q_y(x) : \ y \in Q\} = \inf_t \{t : (x,t) \in \operatorname{convh}\{\operatorname{epi} q_y(x) : \ y \in Q\}\},$$

We prove that $\tilde{f} \in \mathcal{F}_{0,L}(\mathbb{E})$ with $\tilde{f}(x) = f(x)\ \forall x \in Q$. In order to do so, we proceed in the following way:

(a) we prove that $\tilde{f}(x) = f(x)\ \forall x \in Q$,

(b) we show that $\tilde{f}^* \in \mathcal{F}_{1/L,\infty}(\mathbb{E}^*)$ and $\tilde{f} \in \mathcal{F}_{0,\infty}(\mathbb{E})$.

Note that those elements also implies that $\nabla \tilde{f}(x) = \nabla f(x)$ for all $x \in Q$.

We start with (a). As the function is locally smooth on Q, we have

$$f(x) \leq q_y(x), \ \forall x, y \in Q.$$

Therefore, $f(x) \leq \tilde{f}(x)\ \forall x \in Q$. In addition, we have $q_x(x) = f(x)\ \forall x \in Q$ and hence $\tilde{f}(x) \leq f(x)\ \forall x \in Q$. Therefore, $\forall x \in Q$ we have $\tilde{f}(x) \leq f(x) \leq \tilde{f}(x)$ and hence $f(x) = \tilde{f}(x)$.

For proving (b), we use the following identities

$$\tilde{f}^*(s) = \sup_{y \in Q} \left\{q_y^*(s)\right\}, \quad \tilde{f}^{**}(s) = \operatorname{convh}\left\{q_y^{**}(s) : \ y \in Q\right\}.$$

(the proofs of those identities rely on the affine minorization interpretation of conjugation: the convex hull operation corresponds to intersecting the set of affine minorants in the conjugate space; see e.g., [Roc96, Theorem 16.5].)

Since $q_y = q_y^{**}$ (closed convex functions), we have $\tilde{f}^{**} = \tilde{f} \in \mathcal{F}_{0,\infty}(\mathbb{E})$. In order to prove that $\tilde{f}^* \in \mathcal{F}_{1/L,\infty}$, we compute $q_y^*(s)$:

$$\begin{aligned} q_y^*(s) &= \langle s, y\rangle - f(y) + \frac{1}{2L}\|s - \nabla f(y)\|_{\mathbb{E}^*}^2 \\ &= \langle s, y\rangle - \frac{1}{L}\langle s, \nabla f(y)\rangle_{\mathbb{E}^*} - f(y) + \frac{1}{2L}\|\nabla f(y)\|_{\mathbb{E}^*}^2 + \frac{1}{2L}\|s\|_{\mathbb{E}^*}^2. \end{aligned}$$

Hence,

$$\tilde{f}^*(s) = \sup_{y\in Q}\left\{\langle s, y\rangle - \frac{1}{L}\langle s, \nabla f(y)\rangle_{\mathbb{E}^*} - f(y) + \frac{1}{2L}\|\nabla f(y)\|_{\mathbb{E}^*}^2\right\} + \frac{1}{2L}\|s\|_{\mathbb{E}^*}^2,$$

which is $1/L$-strongly convex by Theorem 2.31. Therefore, the statement is proved as $\tilde{f}(x) = f(x)\ \forall x \in Q$ and as $\tilde{f} \in \mathcal{F}_{0,L}(\mathbb{E})$.

□

Remark 2.58. The functional extension proposed in Theorem 2.57 satisfies other definitions of smoothness (see Definition 2.26, Theorem 2.29), as they are equivalent for any smooth convex function.

Note that on the other hand, one can easily prove that a local satisfaction of Definition 2.26 or Theorem 2.29 implies a local satisfaction of Theorem 2.27; therefore, any of those criterion may be checked locally in order to conclude the existence of a smooth convex extension.

Remark 2.59. Local strong convexity is easily handled, as $f(x)+i_Q(x)$ would be a global strongly convex extension to f. In order to obtain a smooth strongly convex extension to some f, one can perform the standard transformation: $h(x) = f(x) - \frac{\mu}{2}\|x\|_{\mathbb{E}}^2$ should be convex and $L-\mu$-smooth on Q. The extension $\tilde{h}(x)$ can then be regularized $\tilde{h}(x)+\frac{\mu}{2}\|x\|_{\mathbb{E}}^2$ in order to obtain a smooth strongly convex extension to f.

The same tip can be used for obtaining a smooth extension to the non-convex locally smooth (on some closed convex set Q) function f: the regularization $h(x) = f(x) + \frac{L}{2}\|x\|_{\mathbb{E}}^2$ is locally convex and smooth, and $\tilde{h}(x) - \frac{L}{2}\|x\|_{\mathbb{E}}^2$ is a globally smooth non-convex extension to f.

Corollary 2.60. Let $f : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$ be closed and proper, and $Q \subseteq \mathbb{E}$ be closed and convex. In addition, let f be differentiable on an open set containing Q, and let f satisfy Conditions (2.4) (smoothness) and (2.7) (strong convexity) on Q. Then, there exists a function $\tilde{f} \in \mathcal{F}_{\mu,L}(\mathbb{E})$ such that $\tilde{f}(x) = f(x)$ and $\nabla\tilde{f}(x) = \nabla f(x)$ for all $x \in Q$.

Corollary 2.61. Let $f : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$ be closed and proper, and $Q \subseteq \mathbb{E}$ be closed and convex. In addition, let f be differentiable on an open set containing Q, and let f satisfy Condition (2.8) (smoothness) on Q. Then, there exists a function $\tilde{f} \in \mathcal{F}_{-L,L}(\mathbb{E})$ such that $\tilde{f}(x) = f(x)$ and $\nabla\tilde{f}(x) = \nabla f(x)$ for all $x \in Q$.

Example 2.62. Consider the function domain $Q = [-1, 1] \subset \mathbb{R}$ and the function $f : \mathbb{R} \to \mathbb{R}$

$$f(x) = \begin{cases} \frac{x^2}{2} & \text{if } |x| \leq 1, \\ -|x| & \text{else.} \end{cases}$$

We note that f is smooth and convex on Q (illustration on Figure 2.6), and its smooth convex extension from Theorem 2.57 is simply $\frac{x^2}{2}$.

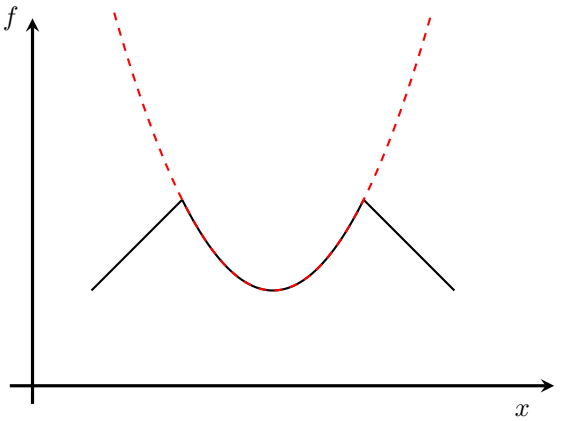


Figure 2.6: Locally smooth convex function f (black), and the smooth convex extension $\tilde{f}$ (dashed, red).

Chapter 3

Convex Interpolation

Throughout this chapter, we study two types of finite representations of convex functions. Also, we provide constructions and geometric interpretations for recovering functions of the desired types from their finite representations.

For a good overview of the chapter, we suggest the reader to thumb through Section 3.1 and Section 3.2 for understanding the main underlying concepts and motivations, and Section 3.3.2 and Section 3.3.1 for the application of the concept to the non-smooth and smooth convex interpolation. The remaining parts can be seen as a collection of similar results for other classes of functions.

The main contributions of the chapter are the following.

- ◇ The concept of convex interpolation, which allows among others to reformulate infinite dimensional optimization problems over those functional spaces (e.g., performance estimation problems) in finite dimension.
- ◇ The derivation of necessary and sufficient conditions for convex interpolation in the cases of different families of convex functions. Among others, we present interpolation conditions for classes of functions involving smoothness and strong convexity. We extend the idea to smooth non-convex functions, to indicator and support functions, and to convex functions with bounded domain or subdifferentials.
- ◇ Extensions of the convex integration problem — and the corresponding cyclic monotonicity conditions — for classes of functions involving smoothness or strong convexity.

This chapter is organized as follows.

- ◇ In Section 3.1, we present the problem and its motivations.
- ◇ Section 3.2 illustrates that commonly used sets of inequalities are generally not sufficient for guaranteeing convex interpolability, especially when

the functions to be interpolated are required to satisfy some smoothness requirements.

- ◇ In Section 3.3, we present the nonsmooth and smooth convex interpolation conditions. We follow a principled approach in order to require the interpolated function to possibly satisfy smoothness, strong convexity and domain and gradient boundedness. Also, we specifically treat the cases of support and indicator functions, and provide an adaptation of the results for handling non-linear smooth functions.
- ◇ Section 3.4 provides the results for the related convex integration problems, and provide a link with former results related to cyclic monotonicity.

Also, note that the subsequent text is based on sections of [THG16b, THG16a].

3.1 Problem and motivations

In this chapter, we study convex functions described by only a finite set of points. The main underlying motivation is the ability to formulate performance estimation problems (which are optimization problems over spaces of functions) in tractable ways (see Chapter 4 and Chapter 5).

Definition 3.1. Let I be a finite index set and $\mathcal{F}$ be a class of convex functions over $\mathbb{E}$, and consider the set of triples $S = \{(x_i, g_i, f_i)\}_{i\in I}$ where $x_i \in \mathbb{E}$, $g_i \in \mathbb{E}^*$ and $f_i \in \mathbb{R}$ for all $i \in I$. The set S is (first-order) $\mathcal{F}(\mathbb{E})$-interpolable if and only if there exists a function $F \in \mathcal{F}$ such that both $g_i \in \partial F(x_i)$ and $F(x_i) = f_i$ hold for all $i \in I$.

The main idea is to develop necessary and sufficient conditions for a set of triplets {(point, gradient, function value)} to be interpolable by a function F within some specified class of (convex) functions $\mathcal{F}$. That is, given a set of triplets $\{(x_i, g_i, f_i)\}_i \subset \mathbb{E}\times\mathbb{E}^*\times\mathbb{R}$ we aim at finding $F \in \mathcal{F}$ such that $f_i = F(x_i)$ and $g_i \in \partial F(x_i)$. One of the main underlying challenges is the incorporation of smoothness constraints into the class of functions $\mathcal{F}$, i.e., to require the existence of a differentiable interpolating function F, i.e., with a Lipschitz condition on its gradient.

In particular cases, it may seem more convenient not to use function values f_i as variables. The problem of recovering a convex function based only on points and gradient values at those points is referred to as the *convex integration* problem.

Definition 3.2. Let I be a finite index set and $\mathcal{F}$ be a class of convex functions, and consider the set of triples $S = \{(x_i, g_i)\}_{i\in I}$ where $x_i \in \mathbb{E}$, $g_i \in \mathbb{E}^*$ for all $i \in I$. The set S is (first-order) $\mathcal{F}(\mathbb{E})$-integrable if and only if there exists a function $F \in \mathcal{F}(\mathbb{E})$ such that $g_i \in \partial F(x_i)$ holds for all $i \in I$.

This problem is related to the so-called *cyclic monotonicity* conditions. Finally, note that subdifferentials are particular cases of *monotone operators*, which we discuss in Section 8.1.

Notations 3.3. We generally refer to $\mathcal{F}$-interpolation or $\mathcal{F}$-integration in the following, without specifying the space it refers to (that is, instead of $\mathcal{F}(\mathbb{E})$ or $\mathcal{F}(\mathbb{E}^*)$), as the corresponding space is generally clear from the context.

3.2 Motivating counterexamples

In this section, we illustrate that the convex interpolation problem is in general not handled using naive approaches, and in particular for requiring the convex interpolated functions to satisfy a smoothness requirement.

For the following counterexamples, we restrict ourselves to the standard Euclidean setting $\mathbb{E} = \mathbb{E}^* = \mathbb{R}^d$ with $\langle x, y\rangle = x^\top y \ \forall x, y \in \mathbb{R}^d$ and $\|.\|_{\mathbb{E}} = \|.\|_{\mathbb{E}^*} = \|.\|_2$ the standard Euclidean 2-norm.

Finding necessary conditions for smooth convex interpolation is relatively easy: starting from any set of conditions that hold over pairs of points from the whole domain of any smooth convex function (for example conditions from Definition 2.26 or Theorem 2.27), one can simply restrict this set to the conditions involving only points x_i with $i \in I$ (i.e., to discretize it). For example, it is well-known that the class of L-smooth convex functions $\mathcal{F}_{0,L}(\mathbb{R}^d)$ is characterized by the pair of inequalities

$$\begin{aligned} &f(y) \geq f(z) + \nabla f(z)^\top (y - z), \qquad \forall\ y, z \in \mathbb{R}^d, \\ &||\nabla f(y) - \nabla f(z)||_2 \leq L||y - z||_2, \qquad \forall\ y, z \in \mathbb{R}^d. \end{aligned} \tag{C1}$$

Therefore, specializing those conditions for $y = x_i$ and $z = x_j$ with $i, j \in I$ leads to the following set of inequalities, which is *necessary* for the existence of an interpolating function in $\mathcal{F}_{0,L}$:

$$\begin{aligned} &f_i \geq f_j + g_j^\top (x_i - x_j), \qquad \forall i, j \in I, \\ &||g_i - g_j||_2 \leq L||x_i - x_j||_2, \qquad \forall i, j \in I. \end{aligned} \tag{C1f}$$

Now, perhaps surprisingly, it turns out that this latter set of conditions is *not sufficient* to guarantee $\mathcal{F}_{0,L}$-interpolability, despite the fact that the originating conditions (C1) *are sufficient* to guarantee that $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$. In order to see that, consider the following example with $I = \{1, 2\}$ and $d = 1$:

$$(x_1, g_1, f_1) = (-1, -2, 1) \text{ and } (x_2, g_2, f_2) = (0, -1, 0).$$

This set satisfies Conditions (C1f) with $L = 1$ but cannot be interpolated by a smooth convex function for any finite value of L: the convexity requirement

forces the interpolating convex function to lie entirely above its linear under-approximations, which lead to an unavoidable non-differentiability at x_1, as illustrated on Figure 3.1. Therefore Conditions (C1f) are not sufficient to guarantee smooth convex interpolation.

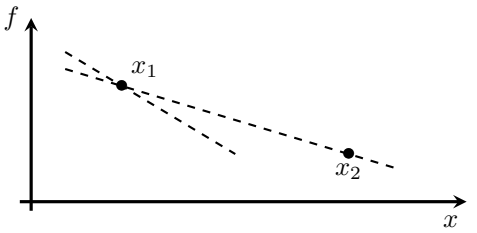


Figure 3.1: Example $(x_1, g_1, f_1) = (-1, -2, 1)$ and $(x_2, g_2, f_2) = (0, -1, 0)$ for $I = \{1, 2\}$ and $d = 1$.

Similarly, we can carry out the same exercise for the following conditions, also well-known to be equivalent to inclusion on $\mathcal{F}_{0,L}(\mathbb{R}^d)$ when imposed on the whole space:

$$\begin{aligned} &f_i \geq f_j + g_j^\top (x_i - x_j), \qquad \forall\ i, j \in I, \\ &f_i \leq f_j + g_j^\top (x_i - x_j) + \frac{L}{2}||x_i - x_j||_2^2, \qquad \forall\ i, j \in I. \end{aligned} \tag{C2f}$$

With an appropriate use of an additional dimension ($d = 2$), one can readily observe that some information may be hidden to this pair of inequalities. Consider the example

$$(x_1, g_1, f_1) = \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, 0 \right) \text{ and } (x_2, g_2, f_2) = \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, 1 \right),$$

from which no smooth convex interpolation can be made (again, unavoidable non-differentiability at both x_1 and x_2). However, both Conditions C1f and C2f are satisfied with $L = 1$.

Those examples illustrate the weakness of a naive approach that consists in discretizing standard necessary and sufficient conditions defined on the whole space. If those discretized conditions were to replace a constraint $f \in \mathcal{F}_{0,L}$ in an optimization problem, they would implicitly allow functions that do not belong to the class $\mathcal{F}_{0,L}$ to be obtained, which would correspond to the solution to a relaxation of the original optimization problem. To conclude this section, note that any set of necessary and sufficient conditions for smooth convex interpolability must be a subset of some necessary and sufficient conditions on the whole domain (since the interpolation conditions have to be satisfied for any discretization of any function within $\mathcal{F}_{0,L}$), whereas the previous examples precisely show that the converse is not true.

In the next subsections, we follow a more principled approach in order to tackle the $\mathcal{F}_{\mu,L}$-interpolation problem. We start with a special case of convex interpolation, that of proper convex functions with no smoothness or strong convexity requirement (i.e., the class $\mathcal{F}_{0,\infty}$).

3.3 Convex interpolation

3.3.1 Non-smooth convex interpolation

We begin by constructing interpolation conditions for the simpler class of non-smooth convex functions $\mathcal{F}_{0,\infty}(\mathbb{E})$. This result is not new but we provide a simple constructive proof of it because it is one of the main building blocks for most following results of this chapter.

Theorem 3.4. The set $\{(x_i, g_i, f_i)\}_{i\in I}$ is $\mathcal{F}_{0,\infty}$-interpolable if and only if

$$f_i \geq f_j + \langle g_j, x_i - x_j\rangle \quad \forall i, j \in I. \tag{3.1}$$

Proof. (Necessity) Assume there exists a convex function $f : \mathbb{E} \to \mathbb{R}$ such that $f_i = f(x_i)$ and $g_i \in \partial f(x_i)$ $\forall i \in I$. The definition of subgradient then immediately implies

$$f_i \geq f_j + \langle g_j, x_i - x_j\rangle \quad \forall i, j \in I.$$

(Sufficiency) Define the following piecewise-linear convex function

$$f(x) = \max_{j\in I} \{f_j + \langle g_j, x - x_j\rangle\}.$$

Since f is the pointwise maximum of a finite number of affine functions, its epigraph is a non-empty polyhedron, and hence f is convex, closed and proper. In addition, $f(x_i) = f_i$ holds by construction. Indeed, we first see that

$$f_i = f_i + \langle g_i, x_i - x_i\rangle, \leq \max_{j\in I} \{f_j + \langle g_j, x_i - x_j\rangle\} = f(x_i).$$

Therefore, we have $f_i \leq f(x_i)$. In addition to this, we have

$$f(x_i) = \max_{j\in I} \{f_j + \langle g_j, x_i - x_j\rangle\} \leq f_i \quad \text{using Condition (3.1) for each } j,$$

which allows to conclude that $f(x_i) = f_i$. The construction also implies that

$g_i \in \partial f(x_i)$ because

$$\begin{aligned} f(x) &= \max_{j \in I} \{f_j + \langle g_j, x - x_j \rangle\} \quad \forall x \in \mathbb{E}, \\ &\geq f_i + \langle g_i, x - x_i \rangle \quad \forall i \in I, x \in \mathbb{E}, \\ &\geq f(x_i) + \langle g_i, x - x_i \rangle \quad \forall i \in I, x \in \mathbb{E}. \end{aligned}$$

□

Remark 3.5. Interpolating functions are typically not unique. Two such interpolating functions are particularly remarkable: one is (pointwise) lower than all the others, and one is (pointwise) higher than all the others. Those two functions naturally arise from the different possibilities for defining a convex function (see Definition 2.8 for the definition in terms of epigraph and Theorem 2.14 for the alternative in terms of subgradients).

Let $\{(x_i, g_i, f_i)\}_{i \in I}$ be satisfying the conditions from Theorem 3.4, we respectively have for the lowest and highest interpolating functions (illustration on Figure 3.2):

$$\begin{aligned} f_l(x) &= \max_{i \in I} \{f_i + \langle g_i, x - x_i \rangle\}, \\ f_h(x) &= \min_{\lambda_i \geq 0} \sum_{i \in I} \lambda_i f_i \quad \text{s.t.} \sum_{i \in I} \lambda_i = 1, \ \sum_{i \in I} \lambda_i x_i = x, \end{aligned}$$

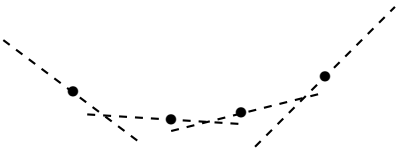
(with the convention $f_h(x) = \infty$ if the previous problem is unfeasible, i.e., if $x \notin \text{conv}\{x_i\}$).

Also, one should note that those interpolation procedures are in fact convex conjugates to each others. For seeing that, we recall that (Theorem 2.21) for a function $f \in \mathcal{F}_{0,\infty}$ we have the following equivalences

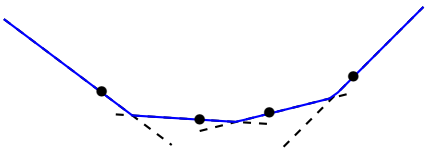
$$g \in \partial f(x) \Leftrightarrow x \in \partial f^*(g) \Leftrightarrow f(x) + f^*(g) = \langle g, x \rangle.$$

Therefore, a function $f \in \mathcal{F}_{0,\infty}$ interpolates the set $S = \{(x_i, g_i, f_i)\}$ if and only if its conjugate $f^* \in \mathcal{F}_{0,\infty}$ interpolates the (conjugate) set $S^* = \{(g_i, x_i, \langle g_i, x_i \rangle - f_i)\}$, and $f = f^{**}$. Hence, $\mathcal{F}_{0,\infty}$-interpolation conditions on S are equivalent to $\mathcal{F}_{0,\infty}$-interpolation of the conjugate set S^*.

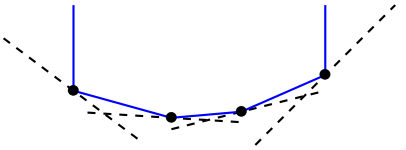
As a conclusion, it can be proved that it is equivalent to interpolate the set S with a construction $f_l : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$ (i.e., the lower bounding function) or to interpolate the set S^* with $f_h : \mathbb{E}^* \to \mathbb{R} \cup \{\infty\}$ (i.e., the upper bounding function) — that is, $f_l = f_h^*$. Note that this can also be understood from Lemma 2.24, as $\forall f, g \in \mathcal{F}_{0,\infty}$, we have $f \geq g \Leftrightarrow g^* \geq f^*$.



(a) Representation of a set of four coordinates, with associated subgradients and function values.



(b) Lower interpolating function for the set provided by Figure 3.2(a). This function takes finite values for every point of $\mathbb{E}$.



(c) Upper interpolating function for the set provided by Figure 3.2(a). The vertical lines on the extreme points of the interval mean that the function takes the value ∞ outside the convex hull of $\{x_i\}_{i\in I}$.

Figure 3.2: Upper and lower interpolating convex functions.

3.3.2 Smooth strongly convex interpolation

We now transform the smooth strongly convex interpolation problem into a convex interpolation one. This is achieved using two previously defined operations: conjugation and minimal curvature subtraction. The reasoning is the following:

(i) Reformulate the $\mathcal{F}_{\mu,L}$ interpolation problem into a $\mathcal{F}_{0,L-\mu}$ interpolation problem using minimal curvature subtraction (Theorem 2.33).

(ii) Write the $\mathcal{F}_{0,L-\mu}$ interpolation problem into a $\mathcal{F}_{1/(L-\mu),\infty}$ interpolation problem using (Legendre-Fenchel) conjugation (Theorem 2.34).

(iii) Transform the $\mathcal{F}_{1/(L-\mu),\infty}$ interpolation problem into a $\mathcal{F}_{0,\infty}$ interpolation problem using again minimal curvature subtraction (Theorem 2.31).

The effect of minimal curvature subtraction on our interpolation problem, used in steps (i) and (iii), is described by the following lemma[1].

[1]In this section, we restrict ourselves to the case $0 \leq \mu < L \leq \infty$ for convenience. However, the results can easily be adapted to the case $L = \mu$. Note that the class $\mathcal{F}_{L,L}(\mathbb{E})$ only contains quadratic functions of the form $f(x) = \frac{L}{2}\|x - c\|_{\mathbb{E}}^2$ for some $c \in \mathbb{E}$.

Lemma 3.6. Consider a set $\{(x_i, g_i, f_i)\}_{i\in I}$ with $x_i \in \mathbb{E}$, $g_i \in \mathbb{E}^*$ and $f_i \in \mathbb{R}$. The following propositions are equivalent for any constants $0 \leq \mu < L \leq +\infty$:

(a) $\{(x_i, g_i, f_i)\}_{i\in I}$ is $\mathcal{F}_{\mu,L}$-interpolable,

(b) $\left\{\left(x_i, g_i - \mu B x_i, f_i - \frac{\mu}{2}\|x_i\|_{\mathbb{E}}^2\right)\right\}_{i\in I}$ is $\mathcal{F}_{0,L-\mu}$-interpolable.

Proof. $[(a) \Rightarrow (b)]$ It follows from Theorem 2.33 that if there exists $f \in \mathcal{F}_{\mu,L}(\mathbb{E})$ interpolating the set, then $h(x) = f(x) - \frac{\mu}{2}\|x\|_{\mathbb{E}}^2$ satisfies $h \in \mathcal{F}_{0,L-\mu}(\mathbb{E})$ and, $\forall i \in I$:

$$h(x_i) = f_i - \frac{\mu}{2}\|x_i\|_{\mathbb{E}}^2, \quad g_i - \mu B x_i \in \partial h(x_i).$$

The set $\left\{\left(x_i, g_i - \mu B x_i, f_i - \frac{\mu}{2}\|x_i\|_{\mathbb{E}}^2\right)\right\}_{i\in I}$ is therefore interpolated by the function $h \in \mathcal{F}_{0,L-\mu}(\mathbb{E})$.

$[(a) \Leftarrow (b)]$ If such a $h \in \mathcal{F}_{0,L-\mu}(\mathbb{E})$ exists and satisfies the interpolation conditions (b), then one can reconstruct a function $f(x) = h(x) + \frac{\mu}{2}\|x\|_{\mathbb{E}}^2$, $f \in \mathcal{F}_{\mu,L}(\mathbb{E})$ which interpolates the set $\{(x_i, g_i, f_i)\}_{i\in I}$. □

The effect of conjugation in step (ii) of the reduction procedure is precisely described in the following lemma.

Lemma 3.7. Consider a set $\{(x_i, g_i, f_i)\}_{i\in I}$ with $x_i \in \mathbb{E}$, $g_i \in \mathbb{E}^*$ and $f_i \in \mathbb{R}$. The following propositions are equivalent $\forall L : \ 0 < L \leq +\infty$:

(a) $\{(x_i, g_i, f_i)\}_{i\in I}$ is $\mathcal{F}_{0,L}$-interpolable,

(b) $\{(g_i, x_i, \langle g_i, x_i\rangle - f_i)\}_{i\in I}$ is $\mathcal{F}_{1/L,\infty}$-interpolable.

Proof. $[(a) \Rightarrow (b)]$ It follows from Theorem 2.34 that if there exists $f \in \mathcal{F}_{0,L}(\mathbb{E})$ then f^* exists and satisfies $f^* \in \mathcal{F}_{1/L,\infty}(\mathbb{E}^*)$. In addition to that, if both f and f^* exists, then they satisfy $\forall i \in I$ the three conditions (see Theorem 2.21):

$$f(x_i) + f^*(g_i) = \langle g_i, x_i\rangle, \quad g_i \in \partial f(x_i), \quad x_i \in \partial f^*(g_i).$$

$[(b) \Rightarrow (a)]$ If a function $f^* \in \mathcal{F}_{1/L,\infty}(\mathbb{E}^*)$ exists and satisfies the interpolation conditions (b), then the conjugate f^{**} (which is convex, proper and closed by construction) satisfies $f^{**} \in \mathcal{F}_{0,L}(\mathbb{E})$ by Theorem 2.34, as well as the interpolation conditions (see Theorem 2.21) $\forall i \in I$:

$$f^{**}(x_i) + f^*(g_i) = \langle g_i, x_i\rangle, \quad g_i \in \partial f^{**}(x_i), \quad x_i \in \partial f^*(g_i).$$

We obtain the desired result by choosing $f = f^{**}$. □

We are now properly armed to define all interpolation equivalences.

Theorem 3.8. The set $\{(x_i, g_i, f_i)\}_{i\in I}$ is $\mathcal{F}_{\mu,L}$-interpolable if and only if the following set of conditions holds for every pair of indices $i \in I$ and $j \in I$

$$f_i - f_j - \langle g_j, x_i - x_j\rangle \geq \frac{1}{2(1-\mu/L)}\left(\frac{1}{L}\|g_i - g_j\|_{\mathbb{E}^*}^2 + \mu\|x_i - x_j\|_{\mathbb{E}}^2 - 2\frac{\mu}{L}\langle g_j - g_i, x_j - x_i\rangle\right). \tag{3.2}$$

Proof. We begin by showing the following equivalences:

(a) $\{(x_i, g_i, f_i)\}_{i\in I}$ is $\mathcal{F}_{\mu,L}$-interpolable,

(b) $\left\{\left(x_i, g_i - \mu B x_i, f_i - \frac{\mu}{2}\|x_i\|_{\mathbb{E}}^2\right)\right\}_{i\in I}$ is $\mathcal{F}_{0,L-\mu}$-interpolable,

(c) $\left\{\left(g_i - \mu B x_i, x_i, \langle g_i, x_i\rangle - f_i - \frac{\mu}{2}\|x_i\|_{\mathbb{E}}^2\right)\right\}_{i\in I}$ is $\mathcal{F}_{1/(L-\mu),\infty}$-interpolable,

(d) $\left\{\left(g_i - \mu B x_i, \frac{Lx_i}{L-\mu} - \frac{B^{-1}g_i}{L-\mu}, \frac{L\langle g_i, x_i\rangle}{L-\mu} - f_i - \frac{\mu L\|x_i\|_{\mathbb{E}}^2}{2(L-\mu)} - \frac{\|g_i\|_{\mathbb{E}^*}^2}{2(L-\mu)}\right)\right\}_{i\in I}$ is $\mathcal{F}_{0,\infty}$-interpolable,

(e) $\left\{\left(\frac{Lx_i}{L-\mu} - \frac{B^{-1}g_i}{L-\mu}, g_i - \mu B x_i, \frac{\mu\langle g_i, x_i\rangle}{L-\mu} + f_i - \frac{\mu L\|x_i\|_{\mathbb{E}}^2}{2(L-\mu)} - \frac{\|g_i\|_{\mathbb{E}^*}^2}{2(L-\mu)}\right)\right\}_{i\in I}$ is $\mathcal{F}_{0,\infty}$-interpolable.

$(a) \Leftrightarrow (b)$ and $(c) \Leftrightarrow (d)$ are direct applications of Lemma 3.6, whereas $(b) \Leftrightarrow (c)$ and $(d) \Leftrightarrow (e)$ are direct applications of Lemma 3.7. Theorem 3.8 follows from equivalence between propositions (a) and (e) applied to the necessary and sufficient conditions for convex interpolation of Theorem 3.4. Finally, it is straightforward to check that condition (e) reduces to the statement of the theorem. □

It is straightforward to establish the equivalent interpolation conditions for both the smooth but non-strongly convex case ($\mu = 0$) and the nonsmooth strongly convex case ($L = +\infty$). In the first case — given by Corollary 3.9 — we find the discrete version of the well-known inequality characterizing L-smooth convex functions, which turns out to be necessary and sufficient:

$$f(x) \geq f(y) + \langle \nabla f(y), x - y\rangle + \frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|_{\mathbb{E}^*}^2.$$

Corollary 3.9. The set $\{(x_i, g_i, f_i)\}_{i\in I}$ is $\mathcal{F}_{0,L}$-interpolable if and only if

$$f_i \geq f_j + \langle g_j, x_i - x_j\rangle + \frac{1}{2L}\|g_i - g_j\|_{\mathbb{E}^*}^2, \qquad \forall i, j \in I.$$

Nonsmooth strongly convex interpolation conditions are given in Corollary 3.10, which corresponds to the well-known inequality characterizing the subgradients of strongly convex functions.

Corollary 3.10. The set $\{(x_i, g_i, f_i)\}_{i\in I}$ is $\mathcal{F}_{\mu,\infty}$-interpolable if and only if

$$f_i \geq f_j + \langle g_j, x_i - x_j\rangle + \frac{\mu}{2}\|x_i - x_j\|_{\mathbb{E}}^2, \qquad \forall i, j \in I.$$

Remark 3.11. Note that one can also easily construct an interpolating function $f(x)$ for the original set of points from Theorem 3.8(a). It follows from Theorem 3.4 that a possible interpolating function for the set $\left\{\left(\tilde{x}_i, \tilde{g}_i, \tilde{f}_i\right)\right\}_{i\in I}$ of Theorem 3.8(c) is given by

$$h(\tilde{x}) = \max_i \left\{ \tilde{f}_i + \langle \tilde{x} - \tilde{x}_i, \tilde{g}\rangle + \frac{1}{2(L-\mu)}\|\tilde{x} - \tilde{x}_i\|_{\mathbb{E}^*}^2 \right\} = \max_i \ h_i(\tilde{x}).$$

This can be conjugated into an interpolating function $h^*(x)$ of the set given by Theorem 3.8(b). Using [Roc96, Theorem 16.5], this can equivalently be written in the form

$$h^*(x) = \operatorname{convh}\left(h_i^*(x)\right),$$

where the $h^*(x)$ is the function whose epigraph is the convex hull of the epigraphs of the h_i^*'s. Hence an interpolating function for the original set $\{(x_i, g_i, f_i)\}_{i\in I}$ is given by

$$f(x) = \operatorname{convh}\left(h_i^*(x)\right) + \frac{\mu}{2}\|x\|_{\mathbb{E}}^2.$$

We provide an example of such an interpolating function on Figure 3.3.

Remark 3.12. Finding interpolation condition involving second-order derivatives seems a lot more challenging. Nevertheless, if such interpolation conditions involving second-order derivatives exist, they should rely on second-order convex analysis [Roc99], in which Fenchel-Legendre conjugation plays a similar role as for first-order derivatives (see e.g., [Cro77]).

Remark 3.13. As in the non-smooth convex case (see Remark 3.5), we provide (pointwise) highest and lowest interpolating functions for smooth convex interpolation. The upper bounding function has already been mentioned as the convex hull of the upper bounding quadratic functions (see Remark 3.11 and Figure 3.3). The value of this function at any point can be obtained by solving

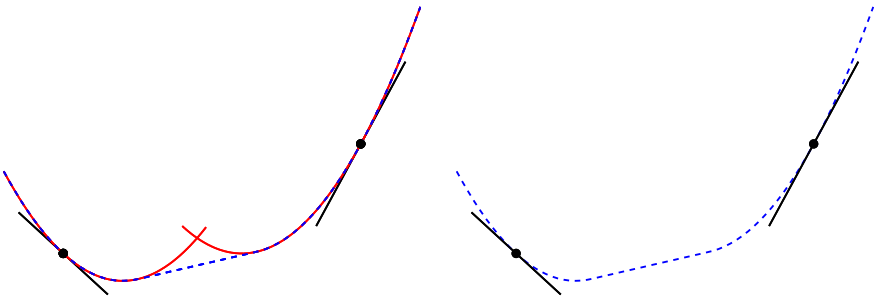


Figure 3.3: Example of an interpolating function; the data triples to be interpolated by a 1-smooth convex function are $(x_1, g_1, f_1) = (2, 2, 3)$ and $(x_2, g_2, f_2) = (-3, -1, 1)$. Figure shows the upper-bounding quadratic functions $h_i^*(x)$ (red, left), the interpolating function $f(x) = \text{convh}\,(h_i^*(x))$ (dashed blue) and the gradients (black tangents).

the following convex quadratically constrained quadratic program[2] (QCQP):

$$\begin{aligned} f_h(x) = \max_{d\in\mathbb{E}^*, y_i\in\mathbb{E}, c\in\mathbb{R}} & \langle d, x\rangle + c, && (3.3)\\ \text{s.t. } & d = LB(y_i - x_i) + g_i && \forall i \in I,\\ & \frac{L}{2}\|y_i\|_{\mathbb{E}}^2 + c + \langle g_i, x_i\rangle - \frac{L}{2}\|x_i\|_{\mathbb{E}}^2 - f_i \le 0 && \forall i \in I, \end{aligned}$$

where the variables are d, c and $\{y_i\}_{i\in I}$.

The interpretation of those variables is the following: d corresponds to the gradient of $f_h(x)$ at x, c corresponds to the value of the intercept of the corresponding linear function to have $f_h(x_i) = \langle d, x\rangle + c$. On the other hand, y_i corresponds to the point where the quadratic upper bound from point x_i has the slope d (required by the set of equality constraints). Finally, the set of inequality constraints requires that the quadratic upper bounds are above $\langle d, x\rangle + c$ at point y_i (and therefore that the quadratic upper bounds are always above the linear function $\langle d, x\rangle + c$).

The lowest interpolating function can be obtained by solving the alternative convex QCQP:

$$\begin{aligned} f_l(x) = \min_{d\in\mathbb{E}^*, y_i\in\mathbb{E}, c\in\mathbb{R}} & \frac{L}{2}\|x\|_{\mathbb{E}}^2 + \langle d, x\rangle + c, && (3.4)\\ \text{s.t. } & d = g_i - LBy_i && \forall i \in I,\\ & \frac{L}{2}\|y_i\|_{\mathbb{E}}^2 - \langle g_i, x_i\rangle + f_i - c \le 0 && \forall i \in I. \end{aligned}$$

[2]This formulation can be further simplified, we provide it in this form for readability.

The variables c, d, y_i have the same interpretation as in the previous case. The main difference with the upper bounding function is that this program focuses on expressing $f_l(x)$ as the lower quadratic function with curvature L which is globally underestimated by all first-order approximations $f_i + \langle g_i, x - x_i\rangle$ — whereas f_u was obtained as the highest linear function which is globally overestimated by all quadratic upper bounds $f_i + \langle g_i, x - x_i\rangle + \frac{L}{2}\|x - x_i\|_{\mathbb{E}}^2$. An example of lower interpolating function is provided on Figure 3.4.

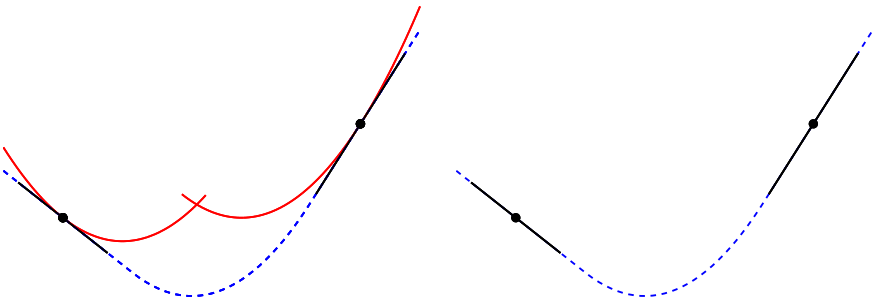


Figure 3.4: Example of a lower interpolating function; the data triples to be interpolated by a 1-smooth convex function are $(x_1, g_1, f_1) = (2, 2, 3)$ and $(x_2, g_2, f_2) = (-3, -1, 1)$. Figure shows the upper-bounding quadratic functions $h_i^*(x)$ (red, left), the interpolating function $f(x) = f_l(x)$ (dashed blue) — see (3.4) — and the gradients (black tangents).

From Remark 3.13, we arrive to the following theorem, which proposes an equivalent and more efficient QCQP for computing the highest and lowest interpolating functions.

Theorem 3.14. Let the set $S = \{(x_i, g_i, f_i)\}_{i\in I}$ be $\mathcal{F}_{\mu,L}$-interpolable. Then, any interpolating function $f \in \mathcal{F}_{\mu,L}$ of S satisfies $f_l(x) \leq f(x) \leq f_h(x)$ for all $x \in \mathbb{E}$, with

$$
\begin{aligned}
f_l(x) = \min_{g\in\mathbb{E}^*, f\in\mathbb{R}} & f, \\
\text{s.t. } & f - f_j - \langle g_j, x - x_j\rangle \geq \frac{1}{2(1-\mu/L)}\left(\frac{1}{L}\|g - g_j\|_{\mathbb{E}^*}^2\right. \\
& \left. +\mu\|x - x_j\|_{\mathbb{E}}^2 - 2\frac{\mu}{L}\langle g_j - g, x_j - x\rangle\right) \qquad \forall j \in I.
\end{aligned}
$$

$$f_u(x) = \min_{g\in\mathbb{E}^*, f\in\mathbb{R}} f,$$
$$\text{s.t. } f_i - f - \langle g, x_i - x\rangle \geq \frac{1}{2(1-\mu/L)}\left(\frac{1}{L}\|g_i - g\|_{\mathbb{E}^*}^2 + \mu\|x_i - x\|_{\mathbb{E}}^2 - 2\frac{\mu}{L}\langle g - g_i, x - x_i\rangle\right) \quad \forall i \in I.$$

In addition, $f_h, f_l \in \mathcal{F}_{\mu,L}$ are themselves interpolating functions for S.

Remark 3.15. Before going into the next section, we provide a simple and intuitive geometric interpretation for the smooth convex interpolation condition from Corollary 3.9

$$f(x) \geq f(y) + \langle \nabla f(y), x - y\rangle + \frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|_{\mathbb{E}^*}^2.$$

Let f be a L-smooth convex function and a given triple $(x, \nabla f(x), f(x))$. We want to find conditions characterizing the possible values for the triplet $(y, \nabla f(y), f(y))$. For doing that, remark that the linear lower bound generated by $(y, \nabla f(y), f(y))$ on f should always be below the quadratic upper bound generated by $(x, \nabla f(x), f(x))$. This condition is the following:

$$f(y) + \langle \nabla f(y), z - y\rangle \leq f(x) + \langle \nabla f(x), z - x\rangle + \frac{L}{2}\|z - x\|_{\mathbb{E}}^2, \ \forall z \in \mathbb{E}.$$

Rewriting this expression, one can obtain the equivalent form

$$0 \leq f(x) - f(y) + \langle \nabla f(y), y - x\rangle + \langle \nabla f(x) - \nabla f(y), z - x\rangle + \frac{L}{2}\|z - x\|_{\mathbb{E}}^2,$$

which should hold for every value of $z \in \mathbb{E}$, and therefore also for the minimum (with respect to z) of the right-hand term:

$$0 \leq f(x) - f(y) + \langle \nabla f(y), y - x\rangle - \frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|_{\mathbb{E}^*}^2,$$

which is exactly the condition from Corollary 3.9.

Example 3.16. Suppose we want to minimize a L-smooth convex function f. We propose a simple geometric interpretation of the interpolation conditions for finding the possible optimal points. We call this the *feasible optimal set*.

Let $S = \{(x_i, \nabla f(x_i), f_i)\}_{i\in I}$ be a set of points which were already evaluated. A new point is possibly optimal for the problem $\min_{x\in\mathbb{E}} f(x)$ if and only if the set $S \cup \{(x, 0, f)\}$ can still be interpolated by a L-smooth convex function (the set S is already $\mathcal{F}_{\mu,L}$-interpolable as it comes from the evaluation of f and its gradient). That is, x is possibly optimal if and only if

$$f_j - \frac{1}{2L}\|\nabla f(x_j)\|_{\mathbb{E}^*}^2 \geq f \geq f_i + \langle \nabla f(x_i), x - x_i\rangle + \frac{1}{2L}\|\nabla f(x_i)\|_{\mathbb{E}^*}^2, \ \forall i, j \in I$$

or equivalently if and only if

$$\min_{j\in I}\{f_j - \frac{1}{2L}\|\nabla f(x_j)\|^2_{\mathbb{E}^*}\} \geq f_i + \langle \nabla f(x_i), x - x_i\rangle + \frac{1}{2L}\|\nabla f(x_i)\|^2_{\mathbb{E}^*}, \forall i \in I.$$

Therefore, given the set S, it is possible that x is optimal if and only if it satisfies the previous *linear* inequalities (illustration on Figure 3.5). Note that there are two possibilities for shrinking the *feasible optimal set*. First, one can diminish the value of the upper bound on the optimal value $f(x_*)$:

$$f(x_*) \leq \min_{j\in I}\{f_j - \frac{1}{2L}\|\nabla f(x_j)\|^2_{\mathbb{E}^*}\},$$

and on the other hand, one can add new points to the set S.

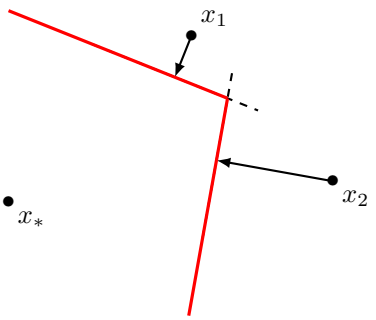


Figure 3.5: Example of feasible optimal set (see Example 3.16). The linear constraint due to x_i is orthogonal to the gradient direction $\nabla f(x_i)$.

3.3.3 Domain and gradient boundedness

In this section, we deal with $\mathcal{S}_{D,\mu}$-interpolability (μ-strongly convex functions with D-bounded domain), which will then serve to obtain $\mathcal{C}_{M,L}$-interpolation (L-smooth convex functions with M-bounded gradients) conditions using Legendre-Fenchel conjugation.

Theorem 3.17 ($\mathcal{S}_{D,\mu}$-interpolability)**.** The set $\{(x_i, g_i, f_i)\}_{i\in I}$ is $\mathcal{S}_{D,\mu}$ (resp. $\mathcal{S}'_{D,\mu}$) interpolable if and only if the following set of conditions holds for every pair of indices $i \in I$ and $j \in I$

$$f_i - f_j - \langle g_j, x_i - x_j\rangle \geq \frac{\mu}{2}\|x_i - x_j\|^2_{\mathbb{E}},$$
$$\|x_j\|_{\mathbb{E}} \leq D \quad (\text{resp. } \|x_j - x_i\|_{\mathbb{E}} \leq D).$$

Proof. (Necessity) Every function $f \in \mathcal{S}_{D,\mu}(\mathbb{E})$ (resp. $f \in \mathcal{S}'_{D,\mu}(\mathbb{E})$) satisfies the conditions.

(Sufficiency) Consider the following construction:

$$f(x) = \begin{cases} \max_i \left\{ f_i + \langle g_i, x - x_i \rangle + \frac{\mu}{2}\|x - x_i\|_{\mathbb{E}}^2 \right\} & \text{if } x \in \operatorname{conv}\left(\{x_i\}_{i\in I}\right) \\ +\infty & \text{else,} \end{cases}$$

One can note that f is μ-strongly convex (convex domain, and maximum of μ-strongly convex functions), and that it indeed interpolates the set $\{(x_i, g_i, f_i)\}_{i\in I}$ (see proof of Theorem 3.4).

Also, we note that $\operatorname{conv}\left(\{x_i\}_{i\in I}\right) \subseteq B_{\mathbb{E}}(0, D)$, with $B_{\mathbb{E}}(0, D)$ the ball of norm $\|.\|_{\mathbb{E}}$ centered at the origin and with radius D. Indeed, choose $z = \sum_{i\in I} \lambda_i x_i$ with $\lambda_i \geq 0$ and $\sum_{i\in I} \lambda_i = 1$, we have

$$\|z\|_{\mathbb{E}} = \left\| \sum_{i\in I} \lambda_i x_i \right\|_{\mathbb{E}} \leq \sum_{i\in I} \lambda_i \|x_i\|_{\mathbb{E}} \leq D,$$

and f has a bounded domain of radius D. Hence $\{(x_i, g_i, f_i)\}_{i\in I}$ is $\mathcal{S}_{D,\mu}$-interpolable, which concludes the proof for the $\mathcal{S}_{D,\mu}$ part. To obtain the same result for $\mathcal{S}'_{D,\mu}$, note that $\forall y, z \in \operatorname{conv}(\{x_i\}_{i\in I})$, we can write $y = \sum_i \lambda_i x_i$ and $z = \sum_i \gamma_i x_i$ with $\lambda_i, \gamma_i \geq 0$ and $\sum_i \lambda_i = \sum_i \gamma_i = 1$. Hence,

$$\begin{aligned} \|y - z\|_{\mathbb{E}} &= \left\| \sum_i \lambda_i (x_i - z) \right\|_{\mathbb{E}} \leq \sum_i \lambda_i \|x_i - z\|_{\mathbb{E}} = \sum_i \lambda_i \left\| \sum_j \gamma_j (x_i - x_j) \right\|_{\mathbb{E}} \\ &\leq \sum_i \lambda_i \sum_j \gamma_j \|x_i - x_j\|_{\mathbb{E}} \leq D. \end{aligned}$$

□

This interpolation result can directly be used for developing interpolation conditions for the class of convex functions with bounded gradient, using the conjugate duality between smoothness and strong convexity on the one hand, and gradient and domain boundedness on the other hand.

Theorem 3.18 ($\mathcal{C}_{M,L}$-interpolability)**.** The set $\{(x_i, g_i, f_i)\}_{i\in I}$ is $\mathcal{C}_{M,L}$ (resp $\mathcal{C}'_{M,L}$) interpolable if and only if the following set of conditions holds for every pair of indices $i \in I$ and $j \in I$

$$f_i - f_j - \langle g_j, x_i - x_j \rangle \geq \frac{1}{2L}\|g_i - g_j\|_{\mathbb{E}^*}^2, \tag{3.5}$$

$$\|g_j\|_{\mathbb{E}^*} \leq M \quad (\text{resp. } \|g_j - g_i\|_{\mathbb{E}^*} \leq M). \tag{3.6}$$

Proof. Note that a function $f \in \mathcal{C}_{M,L}(\mathbb{E})$ (resp. $f \in \mathcal{C}'_{M,L}(\mathbb{E})$) interpolates the set $\{(x_i, g_i, f_i)\}_{i\in I}$ if and only if there exists a corresponding conjugate function $f^* \in \mathcal{S}_{M,1/L}(\mathbb{E}^*)$ (resp. $f^* \in \mathcal{S}'_{M,1/L}(\mathbb{E}^*)$) interpolating the set

$$\{(g_i, x_i, \langle x_i, g_i\rangle - f_i)\}_{i\in I} = \left\{(\tilde{x}_i, \tilde{g}_i, \tilde{f}_i)\right\}_{i\in I}$$

(using exactly the same idea as for Lemma 3.7 along with Corollary 2.44). Using interpolation conditions from Theorem 3.17, such a f^* exists if and only if

$$\begin{aligned}
&\tilde{f}_i - \tilde{f}_j - \langle \tilde{g}_j, \tilde{x}_i - \tilde{x}_j\rangle \geq \frac{1}{2L}\|\tilde{x}_i - \tilde{x}_j\|^2_{\mathbb{E}^*},\\
&\|\tilde{x}_j\|_{\mathbb{E}^*} \leq M \quad (\text{resp. } \|\tilde{x}_j - \tilde{x}_i\|_{\mathbb{E}^*} \leq M),
\end{aligned}$$

which are equivalent to conditions (3.5) and (3.6). □

3.3.4 Indicator and support functions

In this short section, we specifically focus on indicator and support functions. We refer to Section 2.5.3 for definitions and notations.

Indicator functions. Let us now consider the special case of interpolating indicator functions. Basically, this problem is a particular case of the $\mathcal{S}_{D,0}$ (or $\mathcal{S}'_{D,0}$)-interpolation problem. This class of function is particularly interesting for projections in the context of performance estimation.

Theorem 3.19 ($\mathcal{I}_D$-interpolability)**.** The set $\{(x_i, g_i, f_i)\}_{i\in I}$ is $\mathcal{I}_D$ (resp. $\mathcal{I}'_D$) interpolable if and only if the following inequalities hold $\forall i, j \in I$:

$$\begin{aligned}
&f_i = 0,\\
&\langle g_j, x_i - x_j\rangle \leq 0,\\
&\|x_i\|_{\mathbb{E}} \leq D \quad (\text{resp. } \|x_j - x_i\|_{\mathbb{E}} \leq D).
\end{aligned} \tag{3.7}$$

Proof. (Necessity) Any function $f \in \mathcal{I}_D(\mathbb{E})$ (resp. $f \in \mathcal{I}'_D(\mathbb{E})$) satisfies those conditions.

(Sufficiency) Let us construct a convex set whose indicator function interpolate for the set $\{(x_i, g_i, 0)\}$. That is, we construct a closed convex set Q containing all x_i's, and such that $\forall x \in Q$ we have $\langle g_i, x - x_i\rangle \leq 0$ and such that $\|x\|_{\mathbb{E}} \leq D$ $\forall x \in Q$ (resp. $\|x - y\|_{\mathbb{E}} \leq D$ $\forall x, y \in Q$). We start with the simpler case $D = \infty$, by considering the polyhedral set

$$Q = \{x \in \mathbb{E} \mid \langle a_j, x\rangle \leq b_j \ \forall j \in I\},$$

with $a_j = g_j$ and $b_j = \langle g_j, x_j \rangle$. The construction guarantees that $x_i \in Q$. Indeed, by Condition (3.7) we have:

$$\langle g_j, x_i \rangle \leq \langle g_j, x_j \rangle,$$

which is equivalent to $\langle a_j, x_i \rangle \leq b_j$ using the definitions of a_j and b_j, and therefore guarantees that $x_i \in Q$. In order to add the boundedness requirement, we modify the set Q in the following way:

$$\tilde{Q} = Q \cap \operatorname{conv}(\{x_i\}_{i \in I}).$$

This new set is still convex (intersection of two convex sets), it also trivially still satisfies $x_i \in \tilde{Q}$ (which are by construction both contained in Q and $\operatorname{conv}(\{x_i\}_i)$) and $\langle g_i, x - x_i \rangle \leq 0 \ \forall x \in \tilde{Q}$ (since $\tilde{Q} \subseteq Q$). In addition, Q has a radius bounded above by D, because D is an upper bound on the radius (resp. diameter) of $\operatorname{conv}(\{x_i\}_i)$. It is therefore clear that the indicator function $I_{\tilde{Q}} \in \mathcal{I}_D(\mathbb{E})$ (resp. $\mathcal{I}'_D(\mathbb{E})$) interpolates $\{(x_i, g_i, 0)\}_{i \in I}$ as the convex hull has a radius (resp. diameter) D (see proof of Theorem 3.17). □

Support functions. Interpolation conditions for support functions very naturally follows from those for indicator functions (see Section 2.5.3 for more details).

Indeed, requiring a set $S = \{(x_i, g_i, f_i)\}_{i \in I}$ to be $\mathcal{I}^*_M$ (resp. $\mathcal{I}'^*_M$)-interpolable is equivalent to require the set $\tilde{S} = \{(g_i, x_i, \langle x_i, g_i \rangle - f_i)\}_{i \in I}$ to be $\mathcal{I}_M$ (or $\mathcal{I}'_M$)-interpolable.

Corollary 3.20 ($\mathcal{I}^*_M$-interpolability)**.** The set $\{(x_i, g_i, f_i)\}_{i \in I}$ is $\mathcal{I}^*_M$ (resp. $\mathcal{I}'^*_M$)-interpolable if and only if the following inequalities hold $\forall i, j \in I$:

$$\begin{aligned} &\langle g_i, x_i \rangle - f_i = 0, \\ &\langle g_i - g_j, x_j \rangle \leq 0, \\ &\qquad \|g_i\|_{\mathbb{E}^*} \leq M, \quad (\text{resp. } \|g_i - g_j\|_{\mathbb{E}^*} \leq M). \end{aligned}$$

3.3.5 Smooth non-convex interpolation

In this short section, we show how to extend convex interpolation to smooth non-convex interpolation. From Lemma 2.53, it is now straightforward to establish the desired interpolation conditions.

Theorem 3.21. Let $L \in \mathbb{R}^{++}$, the set $\{(x_i, g_i, f_i)\}_{i\in I}$ is $\mathcal{F}_{-L,L}$- interpolable if and only if the following inequality holds $\forall i, j \in I$:

$$f_i \geq f_j - \frac{L}{4}\|x_i - x_j\|_{\mathbb{E}}^2 + \frac{1}{2}\langle g_i + g_j, x_j - x_i\rangle + \frac{1}{4L}\|g_i - g_j\|_{\mathbb{E}^*}^2.$$

Proof. As L is positive and finite, it follows from the equivalence of $\mathcal{F}_{-L,L}$-interpolability of $\{(x_i, g_i, f_i)\}_{i\in I}$ and the $\mathcal{F}_{0,2L}$-interpolability of $\{(x_i, g_i + LBx_i, f_i + \frac{L}{2}\|x_i\|_{\mathbb{E}}^2)\}_{i\in I}$. □

As in the case of smooth convex interpolation, there exists lower and upper bounding interpolating functions (in fact, one can adapt every part of Remark 3.13 to cope with the non-convex case). We provide illustrations of those upper and lower bounding interpolating function on Figure 3.6 and Figure 3.7.

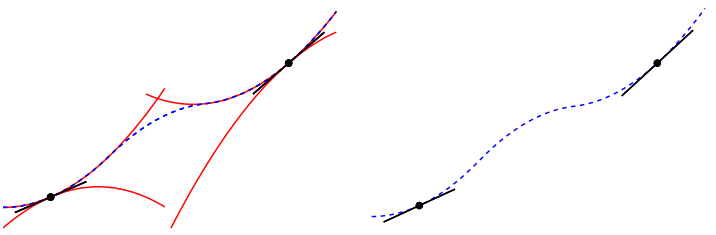


Figure 3.6: Example of an upper interpolating smooth non-convex function; the data triples to be interpolated by a 1/2-smooth function are $(x_1, g_1, f_1) = (2, 1, 2)$ and $(x_2, g_2, f_2) = (-3, -1.25, 0.5)$. Figure shows the upper and lower-bounding quadratic functions (red, left), the interpolating function (dashed blue) and the gradients (black tangents).

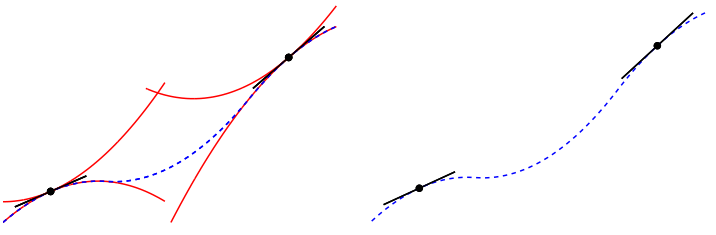


Figure 3.7: Example of a lower interpolating smooth non-convex function; the data triples to be interpolated by a 1/2-smooth function are $(x_1, g_1, f_1) = (2, 1, 2)$ and $(x_2, g_2, f_2) = (-3, -1.25, 0.5)$. Figure shows the upper and lower-bounding quadratic functions (red, left), the interpolating function (dashed blue) and the gradients (black tangents).

3.4 Interpolation without function values

Our interpolation problems are extensions of the classical *finite convex integration problem*, which is concerned with the recovery of a convex function from a set of points x_i, each associated with a subgradient g_i (i.e., function values are not specified). Finite convex integration is treated in details in [LCNS04] in the convex case $\mu = 0$ and $L = \infty$. It is the finite version of the continuous convex integrability problem, which is treated in [Roc96].

A direct necessary and sufficient set of conditions for deciding whether a set $S = \{(x_i, g_i)\}_{i\in I}$ is convex integrable is to require the existence of function values f_i for which the set $\{(x_i, g_i, f_i)\}_{i\in I}$ is convex interpolable (see Theorem 3.4). It is however also possible to derive a set of inequalities that does not involve unknown function values f_i, using so-called *cyclic monotonicity* conditions. However, we will see that those conditions involve a much larger set of inequalities, and that adding the function values as variables to the convex integration problem is a good example of an extended formulation that renders it tractable.

In what follows, we first provide a simple proof for the well-known finite convex integration problem (see e.g., [LCNS04]) and then extend the finite convex integration problem to possibly handle smoothness and strong convexity. On the other hand, domain and gradient boundedness are not treated as they easily follows from the other results.

3.4.1 Rockafellar's cyclic monotonicity

Notations 3.22. Let $S = \{(x_i, g_i)\}_{i\in I}$ for some index set I. For avoiding the pain of using double indices in the sequel, we use the notation $(z, z^*) \in S$ for meaning that $z = x_i$ and $z^* = g_i$ for some $i \in I$.

Definition 3.23. The set $S = \{(x_i, g_i)\}_{i\in I}$ is cyclically monotone if for every cyclic sequence $(z_1, z_1^*), \ldots, (z_m, z_m^*), (z_{m+1}, z_{m+1}^*) \in S$ with $(z_{m+1}, z_{m+1}^*) = (z_1, z_1^*)$, the following (monotonicity) condition is satisfied:

$$\sum_{i=1}^{m} \langle z_i^*, z_{i+1} - z_i \rangle \leq 0.$$

Note that the cyclic monotonicity conditions are satisfied if and only if they are satisfied $\forall m \leq |I|$ (as for any cycle with $m > |I|$, the cycle is composed of at least two consecutive cycles of sizes $m \leq |I|$). The following theorem is a slightly simplified version of an old result by Rockafellar [Roc96, Theorem 24.8].

Theorem 3.24. The set S is $\mathcal{F}_{0,\infty}$-integrable if and only if it is cyclically monotone.

Proof. (Necessity) The necessity part is clear as all monotonicity conditions can be obtained from the definition of subgradient. Indeed, let us chose a cyclic sequence $(z_1, z_1^*), \ldots, (z_m, z_m^*), (z_{m+1}, z_{m+1}^*) \in S$ with $m \geq |I|$ and $z_{m+1} = z_1$. We have that:

$$\begin{aligned} f(z_2) - f(z_1) &\geq \langle z_1^*, z_2 - z_1 \rangle \\ f(z_3) - f(z_2) &\geq \langle z_2^*, z_3 - z_2 \rangle \\ &\vdots \\ f(z_1) - f(z_m) &\geq \langle z_m^*, z_1 - z_m \rangle. \end{aligned}$$

Summing those inequalities produces the monotonicity condition for the cycle:

$$\sum_{i=1}^{m} \langle z_i^*, z_{i+1} - z_i \rangle \leq 0,$$

hence necessity.

(Sufficiency) Assuming the cyclic monotonicity conditions hold, we show that there exists a function $f \in \mathcal{F}_{0,\infty}$ such that $g_i \in \partial f(x_i)\ \forall i \in I$. We construct f in the following way:

$$\begin{aligned} f(x) = \max_{(z_1,z_1^*),\ldots,(z_m,z_m^*)\in S} \big\{ &\langle z_m^*, x - z_m \rangle + \left\langle z_{m-1}^*, z_m - z_{m-1} \right\rangle + \ldots \\ &+ \langle z_1^*, z_2 - z_1 \rangle + \langle g_0, z_1 - x_0 \rangle \big\}, \end{aligned}$$

where $m \geq |I| - 1$[3].

Note that by cyclic monotonicity, this (arbitrary) choice for f implies that $f(x_0) \geq 0$, and by the choice $(z_1, z_1^*) = \ldots = (z_m, z_m^*) = (x_0, g_0)$ we also have that $f(x_0) \leq 0$, and therefore, $f(x_0) = 0$, so f is proper. Also, note that $f \in \mathcal{F}_{0,\infty}$, as it is the maximum of affine functions (i.e., its epigraph is the intersection of epigraphs of closed functions). In order to prove the desired result, it remains to show that $g_i \in \partial f(x_i)\ \forall i \in I$.

By definition of f, we have

$$f(x_i) = \langle z_m^*, x_i - z_m \rangle + \ldots + \langle g_0, z_1 - x_0 \rangle$$

for some $(z_1, z_1^*), \ldots, (z_m, z_m^*) \in S$. Therefore, we can build a global linear underestimate for f, as $\forall y \in E$:

$$\begin{aligned} f(y) &= \max_{(w_1,w_1^*),\ldots,(w_m,w_m^*)\in S} \{ \langle w_m^*, y - w_m \rangle + \ldots + \langle g_0, w_1 - x_0 \rangle \}, \\ &\geq \langle g_i, y - x_i \rangle + \langle z_m^*, x_i - z_m \rangle + \ldots + \langle g_0, z_1 - x_0 \rangle, \end{aligned}$$

[3]More precisely, $m = |I| - 1$ is sufficient, but it does not change much, as one can use different copies of the same x_i among the z_j's.

which shows that $g_i \in \partial f(x_i)$ and allows concluding the proof. □

3.4.2 Smoothness and strong convexity

Using the cyclic monotonicity conditions for convex integration, we note that Theorem 3.8 can also readily be extended to handle the finite (and continuous) integration problems for L-smooth μ-strongly convex functions (i.e., interpolation without function values). Indeed, summing inequality (3.2) from Theorem 3.8 over any cyclic sequence $(z_1, z_1^*), \dots, (z_m, z_m^*), (z_1, z_1^*) \in S$ also produces a necessary inequality that does not involve function values f_i. Moreover, we show in the next section that the set of those inequalities for all possible sequences is necessary and sufficient for finite convex integration of L-smooth μ-strongly convex functions, generalizing the standard cyclic monotonicity conditions. As an illustration, note that the following inequality

$$\langle g_i - g_j, x_i - x_j \rangle \geq \frac{1}{1+\mu/L} \left(\frac{1}{L} \|g_i - g_j\|_{\mathbb{E}^*}^2 + \mu \|x_i - x_j\|_{\mathbb{E}}^2 \right), \tag{3.8}$$

is standard in the analysis of gradient methods on smooth strongly convex functions (see e.g., [Nes04, Theorem 2.1.12]) and corresponds to cycles of length 2. The set of all such inequalities is necessary but not sufficient[4], as it omits longer cycles.

In order to incorporate smoothness (both in the non-convex and (strongly) convex cases), the exact same idea as for non-smooth convex integration can be used, as for other classes of functions. Therefore, we only approach smooth strongly convex integration in what follows.

Lemma 3.25. The set $\{(x_i, g_i)\}_{i \in I}$ is $\mathcal{F}_{\mu,L}$-integrable if and only if the set $\left\{ \left(\frac{Lx_i}{L-\mu} - \frac{B^{-1}g_i}{L-\mu}, g_i - \mu B x_i \right) \right\}_{i \in I}$ is cyclically monotone.

Proof. The following conditions are equivalent (see Theorem 3.8):

(a) $\{(x_i, g_i)\}_{i \in I}$ is $\mathcal{F}_{\mu,L}$-integrable,

(b) $\left\{ \left(\frac{Lx_i}{L-\mu} - \frac{B^{-1}g_i}{L-\mu}, g_i - \mu B x_i \right) \right\}_{i \in I}$ is $\mathcal{F}_{0,\infty}$-integrable.

□

This last theorem allows explicitly formulating the cyclic monotonicity condition for smooth strong convex functions.

[4] A very classical example (for the case $L = \infty$ and $\mu = 0$) is to consider a rotation operator. It satisfies the monotonicity conditions but not the cyclic monotonicity ones. Also, any sampling of points $\{(x_i, g_i)\}$ taken from this operator satisfies the monotonicity condition. Using the transformations from Theorem 3.8, one can adapt this example to be valid for any values $0 \leq \mu < L \leq \infty$.

Theorem 3.26. Let $S = \{(x_i, g_i)\}_{i\in I}$. The set S is $\mathcal{F}_{\mu,L}$-integrable if and only if for any cyclic sequence $(z_1, z_1^*), \dots, (z_m, z_m^*), (z_1, z_1^*) \in S$ we have

$$\sum_{i=1}^{m} \Big[\langle z_i^*, z_i - z_{i+1}\rangle + \frac{1}{L}\langle z_i^*, z_{i+1}^* - z_i^*\rangle_{\mathbb{E}^*} + \mu\langle z_i, z_{i+1} - z_i\rangle_{\mathbb{E}} + \frac{\mu}{L}\big\langle z_{i+1}^* - z_i^*, z_i\big\rangle \Big] \geq 0.$$

For the special values $L = \infty$ and $\mu = 0$ admits as particular cases the smooth convex and strong convex integration results.

Corollary 3.27. The set $S = \{(x_i, g_i)\}$ is $\mathcal{F}_{0,L}$-integrable if and only if for any cyclic sequence $(z_1, z_1^*), \dots, (z_m, z_m^*), (z_{m+1}, z_{m+1}^*) \in S$ with $z_{m+1} = z_1$, we have

$$\sum_{i=1}^{m} \Big[\langle z_i^*, z_i - z_{i+1}\rangle + \frac{1}{L}\langle z_i^*, z_{i+1}^* - z_i^*\rangle_{\mathbb{E}^*} \Big] \geq 0.$$

Corollary 3.28. The set $S = \{(x_i, g_i)\}$ is $\mathcal{F}_{\mu,\infty}$-integrable if and only if for any cyclic sequence $(z_1, z_1^*), \dots, (z_m, z_m^*), (z_{m+1}, z_{m+1}^*) \in S$ with $z_{m+1} = z_1$, we have

$$\sum_{i=1}^{m} \Big[\langle z_i^*, z_i - z_{i+1}\rangle + \mu\langle z_i, z_{i+1} - z_i\rangle_{\mathbb{E}} \Big] \geq 0.$$

Example 3.29. Let us consider the problem of minimizing a L-smooth μ-strongly convex function f. We consider again the problem of characterizing the *feasible optimal region* (subset where it is possible to find the optimal point, see Example 3.16). For doing that, we use the monotonicity condition from Equation (3.8). Also, let $(x_1, \nabla f(x_1))$ be a point we evaluated; a point x is possibly optimal if and only if the set $\{(x_1, \nabla f(x_1)), (x, 0)\}$ is $\mathcal{F}_{\mu,L}$-integrable. This is equivalent to require the following

$$\langle \nabla f(x_1), x_1 - x\rangle \geq \frac{1}{1+\mu/L}\left(\frac{1}{L}\|\nabla f(x_1)\|_{\mathbb{E}^*}^2 + \mu\|x_1 - x\|_{\mathbb{E}}^2\right),$$

which is equivalent to

$$\left\| x - \left(x_1 - \frac{1}{2}\left(\frac{1}{L} + \frac{1}{\mu}\right)B^{-1}\nabla f(x_1)\right)\right\|_{\mathbb{E}}^2 \leq \frac{(L-\mu)^2}{4(L\mu)^2}\|\nabla f(x_1)\|_{\mathbb{E}^*}^2.$$

Hence the optimal point x_* is within a ball of center $x_1 - \frac{1}{2}(\frac{1}{L} + \frac{1}{\mu})B^{-1}\nabla f(x_1)$ and of radius $\frac{(L-\mu)}{2L\mu}\|\nabla f(x_1)\|_{\mathbb{E}^*}$. In addition, this is the exact feasible optimal set under the information provided by only the evaluation of $(x_1, \nabla f(x_1), f(x_1))$.

3.5 Conclusion

In this chapter, we explored two different ways of representing convex and/or differentiable functions in a discrete fashion: with or without function values. Also, we provided procedures for reconstructing the corresponding functions along with corresponding geometrical interpretations.

The following chapters make use of those conditions in order to study the convergence of well-known optimization methods.

Part II

Performance Estimation Problems

Chapter 4

Performance Estimation Problems

The main contributions of the chapter are the following.

- ◇ We provide a methodology for computing the exact worst-case of fixed-step first-order methods for smooth convex (possibly strongly) unconstrained optimization. The methodology relies on reformulating the worst-case computation problem as a convex semidefinite program. As the worst-case computation problem is itself an optimization problem over a class of functions, our convex SDP reformulation relies on using appropriate interpolation conditions (see Chapter 3).
- ◇ We apply our approach to different standard first-order methods, namely the fixed-step gradient method (GM) for smooth (strongly) convex unconstrained optimization, the fast gradient method (FGM) and the optimized gradient method (OGM).

For pedagogical reasons, the approach is described in the simpler case of the standard Euclidean structure $\mathbb{E} = \mathbb{E}^* = \mathbb{R}^d$ with $\langle x, y\rangle = x^\top y \ \forall x, y \in \mathbb{R}^d$ and we also exclusively focus on smooth (strongly) convex objective function in to improve the readability of the chapter. A general approach (allowing among others to consider constraints, regularization terms and different primal and dual Euclidean structures) is presented in Chapter 5.

- ◇ In Section 4.1, we review the concept of performance estimation problem for smooth convex unconstrained optimization. For a short review of the performance estimation history and other recent state-of-the-art methodologies for analyzing first-order optimization methods, we refer to Section 1.3.1.
- ◇ In Section 4.2, we show how the worst-case computation problem can be formulated exactly as a (convex) semidefinite optimization problem,

which provides the first tractable and provably exact formulation of the performance estimation problem. We allow consideration of both smooth convex and smooth strongly convex functions, as well as a large class of performance criteria (a larger class of settings is presented in Chapter 5). We conclude the section with a tight analysis of one iteration of the gradient method (the analysis of one gradient step in the smooth strongly convex case for general step sizes can be found in Appendix 4.A).

- ◇ Section 4.3 then tests our approach numerically on several standard first-order methods for smooth unconstrained minimization, including the constant-step gradient method, the fast gradient method and the optimized gradient method from [KF16d]. We are able to confirm several bounds appearing previously in [DT14], and to conjecture several new worst-case performance bounds, including bounds for strongly convex functions, and bounds on the gradient norm (either for the final iterate, or the smallest norm among all iterates). Another byproduct of our results is a tight estimate of the optimal step size for the gradient method on smooth convex and smooth strongly convex functions.

This chapter is based on sections of the paper [THG16a].

4.1 Introduction to performance estimation

Consider the standard unconstrained minimization problem

$$\min_{x \in \mathbb{R}^d} f(x),$$

where f is a smooth convex function, possibly strongly convex. First-order black-box methods, which only rely on the computation of f and its gradient at a sequence of iterates, can be designed to solve this type of problem iteratively. A central question is then to estimate the accuracy of solutions computed by such a method. More precisely, given a class of problems and a first-order method, one wishes to establish the worst-case accuracy of solutions that can be obtained after applying a given number of iterations, i.e., the performance of the method on the given class of problems.

Many first-order algorithms have been proposed in the literature for smooth convex or smooth strongly convex functions, for which one usually provides a theoretical *upper bound* on the global worst-case accuracy after a number of iterations (see e.g., [Nes04] or [Ber09, Chap.6] for recent overviews). However, many analyses focus on the order of convergence of these bounds, rather than trying to compute exact numerical values. Similarly, *lower bounds* on the performance of first-order black-box methods on given classes of problems can be found in the literature (see e.g., the seminal [NY83]), again often with a focus

on order of convergence. In many situations, the order of convergence of the best available methods match those lower bounds.

Nevertheless, the *exact* numerical value of the worst-case performance of a given method is usually unknown. This is because upper bounds are not assessed precisely, i.e., are known only up to a (possibly unspecified) constant. Another reason is that lower bounds for specific methods are not very frequently developed, and that general lower bounds (valid for all methods) can be quite weak for specific methods, especially if those methods do not feature the best possible order of convergence. Finally, even if exact numerical values are known for both lower and upper bounds, and share the same (optimal) order of convergence, a significant gap between the numerical values of those lower and upper bounds can subsist. If one cares about the worst-case efficiency of a first-order method in practice, this gap can translate into a very large uncertainty on the concrete behavior of a method.

This work is not concerned with orders of convergence. It will focus on the computation of the exact global worst-case performance of a given first-order black-box method, on a given class of functions, after a given number of iterations. We prove that this question can be formulated and solved exactly as a (finite-dimensional) convex optimization problem when the dimension d of the original problem is large, with the following attractive features:

- ◇ Our formulation is a semidefinite optimization problem whose dimension is proportional to the square of the number of iterations of the method to be analyzed.

- ◇ Any dual feasible solution of our formulation provides an upper bound on the worst-case performance. This solution can be easily converted into a standard proof establishing a bound on the performance (i.e., a series of valid inequalities).

- ◇ Any primal feasible solution of our formulation provides a lower bound on the worst case performance. This solution can be easily converted into a concrete function on which the method exhibits the corresponding performance.

- ◇ Hence our formulation is exact, i.e., its optimal value provides the exact worst-case performance.

Our formulation covers both smooth convex functions and smooth strongly convex functions in a unified fashion. It covers a large class of first-order methods which includes the majority of standard methods for smooth unconstrained convex optimization. It can be applied to a variety of performance measures, such as objective function accuracy, gradient norm, or distance to an optimal solution.

4.1.1 Formal definition

Our goal is to express the worst-case performance of an optimization algorithm as the solution of an optimization problem. This approach was pioneered by Drori and Teboulle [DT14], who called it a performance estimation problem (PEP). We now provide a formal definition for this problem.

We consider unconstrained minimization problems involving a given class of objective functions, and only treat first-order black-box methods. This means that the method can only gather information about the objective function using an oracle $\mathcal{O}_f$, which returns first-order information about specific points, i.e., $\mathcal{O}_f(x) = \{f(x), \nabla f(x)\}$. Formally, the first N iterates generated by a first-order black-box method $\mathcal{M}$ (which correspond to N calls of the oracle), starting from an initial point x_0, can be described with

$$\begin{aligned} x_1 &= \mathcal{M}_1\left(x_0, \mathcal{O}_f(x_0)\right), \\ x_2 &= \mathcal{M}_2\left(x_0, \mathcal{O}_f(x_0), \mathcal{O}_f(x_1)\right), \\ &\vdots \\ x_N &= \mathcal{M}_N\left(x_0, \mathcal{O}_f(x_0), \ldots, \mathcal{O}_f(x_{N-1})\right), \end{aligned} \tag{4.1}$$

where $\mathcal{M}_i$ outputs the iterate after the i$^{\text{th}}$ iteration of $\mathcal{M}$.

In order to measure the performance of a given method $\mathcal{M}$ on a specific function f with a specific starting point, we introduce a performance criterion $\mathcal{P}$ to be minimized, that will only depend on the function f and the sequence of the iterates $\{x_0, x_1, \ldots, x_N\}$ generated by the method. Since we are in a black-box setting, we require that the criterion can be computed from the output of the oracle $\mathcal{O}_f$, which has only access to the iterates as well as to an additional point x_*, defined to be any minimizer of function f (the latter being necessary if the criterion has to compare iterates to an optimal solution).

Examples of this performance criterion $\mathcal{P}(\mathcal{O}_f, x_0, \ldots, x_N, x_*)$ include the objective function accuracy $f(x_N) - f(x_*)$, the norm of the gradient $\|\nabla f(x_N)\|$, or the distance to an optimal solution $\|x_N - x_*\|$ (see also Section 4.3.3 for an example of criterion that does not only depend on the last iterate x_N).

Finally, we consider a given class $\mathcal{F}$ of smooth convex or smooth strongly convex functions over $\mathbb{R}^d$, over which we wish to estimate the worst-case performance of a method after N iterations. As we will see in the sequel, specifying the dimension of the class $\mathcal{F}$ to d leads to *two-regime* results: the so-called small scale regime on the one hand (when d is small compared to N), and the large-scale regime on the other one (when d is sufficiently large compared to N).

As methods try to minimize the performance criterion, their worst-case perfor-

mance is obtained by maximizing $\mathcal{P}$ over functions in $\mathcal{F}$, which can be written as

$$\begin{aligned} w(\mathcal{F}, R, \mathcal{M}, N, \mathcal{P}) = \sup_{f, x_0, \ldots, x_N, x_*} \quad & \mathcal{P}(\mathcal{O}_f, x_0, \ldots, x_N, x_*) \\ \text{such that } & f \in \mathcal{F} \\ & x^* \text{ is optimal for } f, \\ & x_1, \ldots, x_N \text{ is generated from } x_0 \text{ by method } \mathcal{M} \text{ with } \mathcal{O}_f, \\ & \|x_0 - x_*\|_2 \leq R. \end{aligned} \tag{PEP}$$

Parameter R was introduced to bound the distance between the initial point x_0 and the optimal solution x_*. Indeed, it is well-known that in most situations, performance of a first-order method cannot be sensibly assessed without such a constraint (see also the discussion of Section 4.2.5).

4.1.2 Finite-dimensional formulation using interpolation

Because it involves an unknown function f as a variable, problem (PEP) is infinite-dimensional. Nevertheless, using the black-box property of the method (and of the performance criterion), we will show that a completely equivalent finite-dimensional problem can readily be formulated by restricting the variable f to the knowledge of the output of its oracle $\mathcal{O}_f$ on the iterates $\{x_0, x_1, \ldots, x_N\}$ and x_*. Indeed, denoting the output of the oracle at each iterate x_i by $\mathcal{O}_f(x_i) = \{f_i, g_i\}$, method $\mathcal{M}$ defined by (4.1) can be equivalently rewritten as

$$\begin{aligned} x_1 &= \mathcal{M}_1\left(x_0, f_0, g_0\right), \\ x_2 &= \mathcal{M}_2\left(x_0, f_0, g_0, f_1, g_1\right), \\ &\vdots \\ x_N &= \mathcal{M}_N\left(x_0, f_0, g_0, \ldots, f_{N-1}, g_{N-1}\right). \end{aligned} \tag{4.2}$$

Now, defining a set $I = \{0, 1, 2, \ldots, N, *\}$ for the indices of the iterates, we can reformulate (PEP) into a problem involving only the iterates $\{x_i\}_{i\in I}$, their function values $\{f_i\}_{i\in I}$ and their gradients $\{g_i\}_{i\in I}$ as (using equivalence between optimality of x_* and constraint $g_* = 0$, as our problem is unconstrained)

$$\begin{aligned} w^f(\mathcal{F}, R, \mathcal{M}, N, \mathcal{P}) = \sup_{\{x_i, g_i, f_i\}_{i\in I}} \quad & \mathcal{P}\left(\{x_i, g_i, f_i\}_{i\in I}\right), \\ \text{such that there exists } & f \in \mathcal{F} \text{ such that } \mathcal{O}_f(x_i) = \{f_i, g_i\}\ \forall i \in I, \\ & g_* = 0, \\ & x_1, \ldots, x_N \text{ is generated from } x_0 \text{ by method } \mathcal{M} \\ & \quad \text{with } \{f_i, g_i\}_{i\in\{0,\ldots,N-1\}}, \\ & \|x_0 - x_*\|_2 \leq R. \end{aligned} \tag{f-PEP}$$

The crucial part of this reformulation is the first constraint, which can be understood as requiring that the set of variables $\{x_i, g_i, f_i\}_{i\in I}$ can be *interpolated* by a function belonging to the class $\mathcal{F}$. This optimization problem is strictly equivalent to the original (PEP) in terms of optimal value, since every solution to (f-PEP) can be interpolated by a solution of (PEP) and, reciprocally, every solution of (PEP) can be discretized to provide a solution to (f-PEP). From that it is clear that $w(\mathcal{F}, R, \mathcal{M}, N, \mathcal{P}) = w^f(\mathcal{F}, R, \mathcal{M}, N, \mathcal{P})$.

4.2 A convex formulation for performance estimation

As explained in the introduction, our performance estimation problem can now be expressed in terms of the iterates and optimal point $\{x_i, g_i, f_i\}_{i\in\{0,\dots,N,*\}}$ only, using the interpolation conditions given by Theorem 3.8.

As our class of functions $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$ and the first-order methods we study are invariant with respect to both an additive shift in the function values and a translation in their domain, we can assume without loss of generality that $x_* = 0$ and $f_* = 0$, which will simplify our derivations. We can also assume $g_* = 0$, from the optimality conditions of unconstrained optimization. The problem can now be stated in its finite-dimensional formulation:

$$w^{(d)}_{\mu,L}(R, \mathcal{M}, N, \mathcal{P}) = \sup_{\{x_i,g_i,f_i\}_{i\in I}\in\left(\mathbb{R}^d\times\mathbb{R}^d\times\mathbb{R}\right)^{N+2}} \mathcal{P}\left(\{x_i, g_i, f_i\}_{i\in I}\right), \quad \text{(d-PEP)}$$

$$\begin{aligned} \text{such that } & \{x_i, g_i, f_i\}_{i\in I} \text{ is } \mathcal{F}_{\mu,L}\text{-interpolable,} \\ & x_1, \dots, x_N \text{ is generated from } x_0 \text{ by method } \mathcal{M} \text{ with (4.2),} \\ & \{x_*, g_*, f_*\} = \{0^d, 0^d, 0\} \text{ and } \|x_0 - x_*\|_2 \leq R. \end{aligned}$$

Problem (d-PEP) is an instance of (f-PEP) where the function class $\mathcal{F}$ is chosen to be $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$, the set of d-dimensional L-smooth μ-strongly convex functions, hence we have $w(\mathcal{F}_{\mu,L}(\mathbb{R}^d), R, \mathcal{M}, N, \mathcal{P}) = w^{(d)}_{\mu,L}(R, \mathcal{M}, N, \mathcal{P})$. Interestingly, in most situations of interest, quantity $w^{(d)}_{\mu,L}(R, \mathcal{M}, N, \mathcal{P})$ is monotonically increasing with d, as a higher dimensional function can usually mimic a lower dimensional one (see Theorem 4.2 and subsequent results for a discussion on finite convergence of this sequence).

Finally, note that problem (d-PEP) is not convex, as it involves several non-convex quadratic constraints (e.g., $g_j^\top x_i$ terms in the interpolation conditions). In the next section, we show how (d-PEP) can be cast as a convex semidefinite program [VB94] when dealing with a certain class of first-order black-box methods, those based on fixed steps.

4.2.1 Fixed-step first-order methods

We hereby restrict ourselves to the class of fixed-step first-order methods, where each iterate is obtained by adding a series of gradients steps with fixed step sizes to the starting point x_0.

Definition 4.1. A method $\mathcal{M}$ is called a *fixed-step method* if its iterates are computed according to

$$x_i = x_0 - \sum_{k=0}^{i-1} h_{i,k} g_k.$$

with fixed scalar coefficients $h_{i,k}$.

A fixed-step method performing N steps is completely defined by the lower triangular $N \times N$ matrix $H = \{h_{i,k}\}_{1 \leq i \leq N, 0 \leq k \leq N-1}$ (where $h_{i,k}$ is defined to be zero if $k \geq i$). Many classical methods such as the gradient method with constant step size (GM) and the fast gradient method (FGM) are included in this class of algorithms (see the details in Section 4.3).

4.2.2 A convex reformulation using a Gram matrix

In order to obtain a convex formulation for (d-PEP), we introduce a Gram matrix[1] to describe the iterates and their gradients. Denoting

$$P = [g_0 \ g_1 \ \dots \ g_N \ x_0]$$

we define the symmetric $(N+2) \times (N+2)$ Gram matrix $G = P^\top P \in \mathbb{S}^{N+2}$, which is equivalent to

$$G = \{G_{i,j}\}_{0 \leq i,j \leq N} \text{ with } \begin{cases} G_{i,j} = g_i^\top g_j & \text{for any } 0 \leq i,j \leq N, \\ G_{N+1,j} = x_0^\top g_j & \text{for any } 0 \leq j \leq N, \\ G_{i,N+1} = g_i^\top x_0 & \text{for any } 0 \leq i \leq N, \\ G_{N+1,N+1} = x_0^\top x_0 \end{cases}$$

(note that the size of this matrix does not depend on the dimension of iterate x_0 and gradients g_i).

The constraints in problem (d-PEP) can now be entirely formulated in terms of the entries of the Gram matrix G along with the function values f_i. Indeed all iterates apart from x_0 can be substituted out of the formulation using Definition 4.1 of a fixed-step method, and the resulting formulation only involves function values f_i and inner products between x_0 and all gradients g_i.

[1] This sort of lifted representation was made famous among others for the MAX-CUT problem [GW95].

Note that the initial iterate x_0 and successive gradients g_i of any solution to problem (d-PEP) can be transformed into a symmetric and positive semidefinite Gram matrix G. Moreover, since vectors x_0 and g_i belong to $\mathbb{R}^d$, matrix G has rank at most d. In the other direction, it is easy to see that any symmetric and positive semidefinite Gram matrix G of rank at most d can be converted back (using Cholesky decomposition for example) into $N+2$ vectors $x_0 \in \mathbb{R}^d$ and $g_i \in \mathbb{R}^d$ which describe the initial iterate and successive gradients of a d-dimensional function (this transformation is however not unique). From those observations we can anticipate that an equivalent formulation of (d-PEP) will rely on imposing that G is symmetric and positive semidefinite, which is a convex constraint and will naturally lead to a semidefinite program.

4.2.3 Exact worst-case performance of fixed-step first-order methods as a semidefinite program

For notational convenience, we define vectors $h_i \in \mathbb{R}^{N+2}$ for any i between 0 and N and $h_* \in \mathbb{R}^{N+2}$ as follows (see Definition 4.1)

$$h_i^\top = [-h_{i,0} \ -h_{i,1} \ \dots \ -h_{i,i-1} \ 0 \ \dots \ 0 \ 1], \quad h_*^\top = [0 \ \dots \ 0],$$

so that we have $x_i = Ph_i$. In order to lighten the notations we also define $u_i = e_{i+1} \in \mathbb{R}^{N+2}$, the canonical basis vectors, and u_* the vector of zeros. Using those notations, we rewrite the interpolation constraints (3.2) from Theorem 3.8 in the following form for all $i, j \in I$:

$$\begin{aligned} f_i \geq & f_j + \frac{L}{L-\mu}(u_j^\top G h_i - u_j^\top G h_j) + \frac{1}{2(L-\mu)}(u_i - u_j)^\top G (u_i - u_j) \\ & + \frac{\mu}{L-\mu}(u_i^\top G h_j - u_i^\top G h_i) + \frac{L\mu}{2(L-\mu)}(h_i - h_j)^\top G (h_i - h_j). \end{aligned}$$

We can equivalently formulate all constraints using the trace operator, and add the distance constraint $\|x_0 - x_*\|_2 \leq R$ on the starting point as well as the positive semidefiniteness constraint for G. Defining matrices A_{ij} and A_R in the following way for all $i, j \in I$:

$$\begin{aligned} 2A_{ij} = & \frac{L}{L-\mu}\left(u_j(h_i - h_j)^\top + (h_i - h_j)u_j^\top\right) + \frac{1}{L-\mu}(u_i - u_j)(u_i - u_j)^\top \\ & + \frac{\mu}{L-\mu}\left(u_i(h_j - h_i)^\top + (h_j - h_i)u_i^\top\right) + \frac{L\mu}{L-\mu}(h_i - h_j)(h_i - h_j)^\top, \\ A_R = & \, u_{N+1}u_{N+1}^\top. \end{aligned}$$

We obtain the following compact formulation for the feasible region that is *linear* in its variables $f \in \mathbb{R}^{N+1}$ and $G \in \mathbb{S}^{N+2}$

$$\begin{aligned} f_j - f_i + \operatorname{Tr}(GA_{ij}) &\leq 0, \qquad \text{for all } i,j \in I, \\ \operatorname{Tr}(GA_R) - R^2 &\leq 0, \\ G &\succeq 0\,, \end{aligned}$$

with an additional non-convex rank constraint

$$\operatorname{rank}(G) \leq d,$$

for imposing the dimension of the original problem.

From the discussion at the end of the previous section, it is easy to see that any d-dimensional function f and starting point $x_0 \in \mathbb{R}^d$ produce a feasible solution (f, G) where matrix G has rank at most d. On the other hand, any feasible solution (f, G) where G has rank at most d can be interpolated into a d-dimensional function $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ and a starting point $x_0 \in \mathbb{R}^d$. Indeed, matrix $G = P^\top P$ provides $x_0 \in \mathbb{R}^d$ and $N+1$ successive gradients $g_i \in \mathbb{R}^d$, while the other iterates x_i derive from the definition of the method. Our interpolating conditions ensure that a function compatible with these data triples $\{x_i, g_i, f_i\}_{i \in I}$ exists.

Considering finally the performance criterion $\mathcal{P}$, we observe that any concave semidefinite-representable function in G and f leads to a worst-case estimation problem that can be cast as a convex semidefinite optimization problem (see e.g., [BTN01]) plus a rank constraint. In particular, linear functions of the entries of f and G are suitable. Classical performance criteria such as $f(x_N) - f_*$, $\|\nabla f(x_N)\|_2^2$ and $\|x_N - x_*\|_2^2$ are indeed covered by this formulation. We focus below on the case of a linear performance criterion, but note that other criteria can be useful (see for example a concave piecewise linear criteria used in Section 4.3.3).

We can now state the main result of this chapter.

Theorem 4.2. Consider the class $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$ of L-smooth μ-strongly convex functions with $0 \leq \mu < L \leq \infty$, a fixed-step first-order method that computes N iterates according to matrix $H \in \mathbb{R}^{N \times N}$, and a performance criterion $\mathcal{P}_{b,C}(f, G) = b^\top f + \operatorname{Tr}(CG)$ that depends linearly on the function values at those iterates and quadratically on the iterates and their gradients ($b \in \mathbb{R}^{N+1}$ and $C \in \mathbb{S}^{N+2}$).

The worst-case performance after N iterations of method H applied to some function in $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$ is equal to the optimal value of the following rank-

constrained semidefinite program

$$\begin{aligned} w^{(d)}_{\mu,L}(R,H,N,b,C) = \sup_{G\in\mathbb{S}^{N+2},f\in\mathbb{R}^{N+1}} \quad & b^\top f + \operatorname{Tr}(CG) && \text{(sdp-PEP(d))}\\ \text{such that } & f_j - f_i + \operatorname{Tr}(GA_{ij}) \leq 0, && i,j\in I,\\ & \operatorname{Tr}(GA_R) - R^2 \leq 0,\\ & G \succeq 0,\\ & \operatorname{rank}(G) \leq d. \end{aligned}$$

Alternatively, if $N \leq d-2$, the worst-case performance after N iterations of method H applied to some function in $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$ is equal to the optimal value of the following convex semidefinite program

$$\begin{aligned} w^{\text{sdp}}_{\mu,L}(R,H,N,b,C) = \sup_{G\in\mathbb{S}^{N+2},f\in\mathbb{R}^{N+1}} \quad & b^\top f + \operatorname{Tr}(CG) && \text{(sdp-PEP)}\\ \text{such that } & f_j - f_i + \operatorname{Tr}(GA_{ij}) \leq 0, && i,j\in I,\\ & \operatorname{Tr}(GA_R) - R^2 \leq 0,\\ & G \succeq 0, \end{aligned}$$

with matrices A_{ij} and A_R as defined above. In others words,

$$w^{\text{sdp}}_{\mu,L}(R,H,N,b,C) = w^{(d)}_{\mu,L}(R,H,N,b,C) \text{ for any } d \geq N+2.$$

Proof. We have already shown the two-way correspondence between functions in $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$ and feasible solutions of this problem where matrix G has rank at most d. Since matrix G has size $N+2$, it has rank at most $N+2$, which establishes that this semidefinite program is a correct formulation of the performance estimation problem when $d \geq N+2$. □

The optimal value $w^{\text{sdp}}_{\mu,L}(R,H,N,b,C)$ of (sdp-PEP(d)) is not necessarily finite or attained at some feasible point. However, when L is finite, any continuous performance criterion $\mathcal{P}$ will force the optimal value to be attained and finite.

Proposition 4.3. Under the assumptions of Theorem 4.2, the optimum value of (sdp-PEP(d)) is attained and finite when $L < \infty$.

Proof. To show that the solution of (sdp-PEP(d)) is attained and finite, it suffices to prove that its feasible region is compact (since the objective is continuous). We first prove that the iterates of method H applied to any function in $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$ with $L < \infty$ are bounded, as well as their gradients.

Note that the Lipschitz condition on the gradients (C1f) with $j = *$ shows that if iterate x_i is bounded, gradient g_i is also bounded. We proceed by recurrence. We start with the fact that x_0 is bounded, using the assumption that $x_* = 0$ and constraint $\|x_0 - x_*\|_2 \leq R$. This implies that g_0 is bounded, hence that

x_1 is bounded using Definition 4.1 of a fixed-step method. This implies in turn that g_1 is bounded, then that x_2 is bounded, and so on until we have shown that all iterates and gradients are bounded.

Condition (C2f) with $j = *$ then implies that function values f_i are bounded. Therefore all entries in variables f and G are bounded which, combined with closeness of the feasible region, establishes the claim. □

Remark 4.4. When $L = +\infty$ (recall the conventions $1/+\infty = 0$ and $+\infty - \mu = +\infty$ used in this chapter), the feasible region may be unbounded and it is possible to design feasible functions which drive standard performance criteria arbitrarily away from 0. Nevertheless, performance estimation on such nonsmooth functions could still be tackled after introduction of another appropriate Lipschitz condition on the class of functions, such as $||g_i||_2 \leq L$. We leave this as a topic for further research and, in the rest of this chapter, restrict ourselves to the smooth case $L < +\infty$. See Section 5.3 for examples with $L = +\infty$.

Our formulation (sdp-PEP) is dimension-independent (i.e., it does not depend on the value of d), and computes the exact worst-case performance of a first-order method with N steps as long as the class of functions of interest contains functions of dimension at least $N + 2$. This corresponds to the so-called large-scale optimization setting, which is usually assumed when analyzing the worst-case of first-order methods.

Using the structure of (sdp-PEP(d)), it is straightforward to establish that the sequence $\{w^{(d)}_{\mu,L}(R, H, N, b, C)\}_{d=1,2,\ldots}$ is monotonically increasing, and that it converges for a finite value of d.

Corollary 4.5. The worst-case performance after N steps of a fixed-step method on a L-smooth (μ-strongly) convex function is achieved by an $N + 2$-dimensional function.

Finally, note that, when applied to the gradient method in the non-strongly convex case ($\mu = 0$), problem (sdp-PEP) is equivalent to one of the formulations proposed by Drori and Teboulle in [DT14], more specifically to their problem (G). Theorem 4.2 establishes that this relaxation is in fact exact under large-scale assumptions. In addition, note that the large-scale assumption $N \leq d - 2$ may be conservative, as there may exist low-rank solutions to the SDP.

Remark 4.6. Existence of low-rank solutions in semidefinite programming is an important issue which is addressed in numerous references. As examples, the seminal [Pat98, Bar01] discuss the existence of low-rank solutions depending on the number of constraints appearing in the SDP. More precisely, for a feasible SDP with m affine constraints, there exists an optimal solution of rank at most $\lfloor \frac{\sqrt{8m+1}-1}{2} \rfloor$. As far as we know, those results do not yield interesting conclusions in the case of our general formulation sdp-PEP (which has a quadratic number of constraints). However, other frameworks for studying the properties

of the solutions of (sdp-PEP) using its structure could potentially be of application here, for example existence of low-rank solutions due to graph structures (see e.g.,[LV14]), or dimensionality reduction using lower-dimensional matrix algebra structures (see e.g.,[DK10]). We leave further investigations in those directions for future research.

4.2.4 A dual SDP to generate upper bounds

In general, it is not easy to find an analytical optimal solution to (sdp-PEP). Hence, we are also interested in a generic and easier way of obtaining analytical upper bounds on the performance of a given algorithm. A classical way of doing so is to work with the Lagrangian dual of (sdp-PEP):

$$\begin{aligned} \inf_{\lambda_{ij},\tau} \ \tau R^2 \ \text{ such that } \ & \tau A_R - C + \sum_{i,j\in I} \lambda_{ij} A_{ij} \succeq 0, && \text{(d-sdp-PEP)} \\ & b - \sum_{i,j\in I} \lambda_{ij}(u_j - u_i) = 0, \\ & \lambda_{ij} \geq 0, \qquad i,j \in I, \\ & \tau \geq 0, \end{aligned}$$

whose feasible solutions will provide theoretical upper bounds on the worst-case behavior of every fixed-step first-order method (using weak duality). Note that the final dual formulation used in [DT14], which deals with the case $\mu = 0$, can be recovered by taking $\lambda_{ij} = 0$ for $i + 1 \neq j$ or $i \neq *$ in our dual, i.e., it is a restriction of (d-sdp-PEP) with a potentially larger optimal value.

The next theorem guarantees that no duality gap occurs between (sdp-PEP) and (d-sdp-PEP) under the technical assumption $h_{i,i-1} \neq 0$ ($i \in \{1,\dots,N\}$). This assumption is reasonable as it only implies that, at each iteration, the most recent gradient obtained from the oracle has to be used in the computation of the next iterate. The theorem will also guarantee the existence of a dual feasible point attaining the optimal value of the primal-dual pair of estimation problems (sdp-PEP) and (d-sdp-PEP), i.e., a tight upper bound on the worst-case performance of the considered method.

Theorem 4.7. The optimal value of the dual problem (d-sdp-PEP) with $0 \leq \mu < L < \infty$ is attained and equal to $w^{\text{sdp}}_{\mu,L}(R,H,N,b,C)$ under the assumptions that $h_{i,i-1} \neq 0$ for all $i \in \{1,\dots,N\}$.

Proof. We use the classical Slater condition [BV04] on the primal problem in order to guarantee a zero duality gap — that is, we show that (sdp-PEP) has a feasible point with $G \succ 0$. The reasoning is divided in two parts; we consider first the case $\mu = 0$ and $L = 2+2\cos(\pi/(N+2))$, and we generalize it to general $\mu < L$ afterwards. Consider the quadratic function $f(x) = \frac{1}{2}x^\top Q x$ with the

following tridiagonal positive definite matrix Q

$$Q = \begin{pmatrix} 2 & 1 & 0 & 0 & \dots & 0 \\ 1 & 2 & 1 & 0 & \dots & 0 \\ 0 & 1 & 2 & 1 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \end{pmatrix} \succ 0.$$

We show how to construct a full-rank G feasible for (sdp-PEP) using the values of the quadratic function f. In order to do so, we exhibit a full-rank matrix

$$P = [x_0 \ g_0 \ g_1 \ \dots g_N]$$

corresponding to the application of a given method (with $h_{i,i-1} \neq 0$) to the quadratic function f. Indeed, choosing $x_0 = Re_1$, we can show that P is upper triangular with non-zero diagonal entries. Then we have

$$\begin{aligned} g_0 &= Qx_0 = 2e_1 + e_2, \\ x_1 &= x_0 - h_{1,0}g_0, \\ g_1 &= Qx_1 = g_0 - h_{1,0}Qg_0 = 2e_1 + e_2 - h_{1,0}(4e_1 + 4e_2 + e_3). \end{aligned}$$

Hence g_1 has a non-zero element associated with e_3 whereas the only non-zero elements of g_0 are associated with e_1 and e_2. Now, we assume that g_{i-1} has a non-zero element corresponding to e_{i+1} and zero elements corresponding to e_k for all $k > i+1$, while all previous gradients have zero components corresponding to e_k for all $k > i$. Then we have

$$g_i^\top e_{i+2} = x_i^\top Q e_{i+2} = x_i^\top (e_{i+1} + 2e_{i+2} + e_{i+3}),$$

with $x_i^\top e_{i+2} = x_i^\top e_{i+3} = 0$ and $x_i^\top e_{i+1} \neq 0$ because of the recurrence assumption and the iterative form of the algorithm:

$$\begin{aligned} x_i^\top e_{i+1} &= \underbrace{x_0^\top e_{i+1}}_{=0} - \sum_{k=0}^{i-2} h_{i,k} \underbrace{g_k^\top e_{i+1}}_{=0} - h_{i,i-1} \underbrace{g_{i-1}^\top e_{i+1}}_{\neq 0}, \\ x_i^\top e_{i+2} &= \underbrace{x_0^\top e_{i+2}}_{=0} - \sum_{k=0}^{i-2} h_{i,k} \underbrace{g_k^\top e_{i+2}}_{=0} - h_{i,i-1} \underbrace{g_{i-1}^\top e_{i+2}}_{=0}, \\ x_i^\top e_{i+3} &= \underbrace{x_0^\top e_{i+3}}_{=0} - \sum_{k=0}^{i-2} h_{i,k} \underbrace{g_k^\top e_{i+3}}_{=0} - h_{i,i-1} \underbrace{g_{i-1}^\top e_{i+3}}_{=0}. \end{aligned}$$

Hence, g_i has a non-zero element associated with e_{i+2}. We deduce that the

following components are equal to zero by computing $g_i^\top e_{i+2+k}$ for $k > 0$:

$$g_i^\top e_{i+2+k} = x_i^\top Q e_{i+2+k} = x_i^\top (e_{i+1+k} + 2e_{i+2+k} + e_{i+3+k}),$$

which is zero because of the algorithmic structure of x_i, i.e.,

$$x_i^\top e_{i+1+k} = \underbrace{x_0^\top e_{i+1+k}}_{=0} - \sum_{k=0}^{i-2} h_{i,k} \underbrace{g_k^\top e_{i+1+k}}_{=0} - h_{i,i-1} \underbrace{g_{i-1}^\top e_{i+1+k}}_{=0}.$$

Hence, P is an upper triangular matrix with positive entries on the diagonal, and is therefore full-rank. In order to make this statement hold for general $\mu < L$, observe that the structure of the matrix is preserved using the operation (I_{N+2} is the identity matrix)

$$Q' = (Q - \lambda_{\min}(Q) I_{N+2}) \frac{(L-\mu)}{\lambda_{\max}(Q) - \lambda_{\min}(Q)} + \mu I_{N+2}.$$

The corresponding quadratic function is easily seen to be L-smooth and μ-strongly convex. Therefore, the interior of the domain of (sdp-PEP) is non-empty and Slater's condition applies for $\mu < L$, ensuring that no duality gap occurs and that the dual optimal value is attained. □

One can note that Theorem 4.7 guarantees the existence of a fully explicit proof (i.e., a combination of valid inequalities, or equivalently, a dual feasible solution) for any worst-case function (see the example at the end of this section).

4.2.5 Homogeneity of the optimal values with respect to L and R

We observe that, for most performance criteria, one can predict how the worst-case performance depends from parameters L and R, provided the fixed step sizes contained in H are scaled appropriately (i.e., inversely proportional to L). In the rest of this chapter we will only consider such scaled (normalized) step sizes. Therefore, the corresponding performance estimation problems have only to be solved numerically in the case $R = 1$ and $L = 1$, from which a general bound valid for any L and R can be deduced.

More specifically, a classical reasoning involving appropriate scaling operations easily leads to the following homogeneity relations for the standard criteria $f(x_N) - f_*$, $\|\nabla f(x_N)\|_2$ and $\|x_N - x_*\|_2$ (a proof of the first relation is provided

hereafter):

$$
\begin{aligned}
w^{(d)}_{\mu,L}(R, H/L, N, f(x_N) - f_*) &= LR^2\ w^{(d)}_{\kappa,1}(1, H, N, f(x_N) - f_*),\\
w^{(d)}_{\mu,L}(R, H/L, N, \|\nabla f(x_N)\|_2) &= LR\ w^{(d)}_{\kappa,1}(1, H, N, \|\nabla f(x_N)\|_2),\\
w^{(d)}_{\mu,L}(R, H/L, N, \|x_N - x_*\|_2) &= R\ w^{(d)}_{\kappa,1}(1, H, N, \|x_N - x_*\|_2),
\end{aligned}
$$

where $\kappa = \mu/L$ is the inverse condition number and H/L describes the fixed-step method obtained by dividing all step sizes $h_{i,j}$ by the Lipschitz constant L. Results in the rest of this chapter implicitly rely on these relations.

Proof. Let us provide a proof for the relation

$$
w^{(d)}_{\mu,L}(R, H/L, N, f(x_N) - f_*) = LR^2\ w^{(d)}_{\kappa,1}(1, H, N, f(x_N) - f_*)
$$

to serve as an example. For doing that, we start by defining a constant $\alpha > 0$ and two scaling operations $A_1(\alpha), A_2(\alpha) : \mathcal{F}_{0,\infty}(\mathbb{R}^d) \to \mathcal{F}_{0,\infty}(\mathbb{R}^d)$:

$$
A_1(\alpha) : f(x) \to \alpha f\left(x\right), \qquad\qquad A_2(\alpha) : f(x) \to \alpha^2 f\left(\frac{x}{\alpha}\right).
$$

We make the following observations.

- ◇ First, for all $f \in \mathcal{F}_{\mu,L}$, we have:

$$
A_1(\alpha)\left[f\right] \in \mathcal{F}_{\alpha\mu,\alpha L}, \qquad\qquad A_2(\alpha)\left[f\right] \in \mathcal{F}_{\mu,L}.
$$

- ◇ Second, if $x_1, \ldots, x_N$ are the iterates obtained by a fixed-step first-order method started at x_0 on $f \in \mathcal{F}_{\mu,L}$, then the same iterates are generated from x_0 by the same (scaled) algorithm (i.e., whose coefficients are given by $H/(\alpha L)$) on $f_1 = A_1(\alpha)\left[f\right]$. Hence, we have $f_1(x_N) = \alpha f(x_N)$ and $f_1(x_N) - f_1(x_*) = \alpha(f(x_N) - f(x_*))$, with x_* being optimal for f and f_1.
- ◇ Similarly to the second observation, if $x_1, \ldots, x_N$ are the iterates obtained by a fixed-step first-order method starting from x_0 on $f \in \mathcal{F}_{\mu,L}$, then the iterates $\alpha x_1, \ldots, \alpha x_N$ are generated by the same algorithm on $f_2 = A_2(\alpha)\left[f\right]$ from αx_0. Therefore, we have $f_2(\alpha x_N) = \alpha^2 f(x_N)$ and $f_2(\alpha x_N) - f_2(\alpha x_*) = \alpha^2(f(x_N) - f(x_*))$, with x_* being optimal for f and αx_* for f_2.

In order to conclude, assume that $f \in \mathcal{F}_{\mu,L}$ generate the worst-case value of $f(x_N) - f_*$ for a fixed-step first-order method, for some values of R, L and μ. Then $f_1 = A_1(\alpha)\left[f\right]$ must generate the worst-case value for the same criterion with R, αL and $\alpha\mu$. Indeed, if it was not the case, we could choose the function reaching the worst-case of those new parameters (R, αL and $\alpha\mu$) and use the application $A_1(\alpha^{-1})$ in order to generate something worse than f in the original setting, which would contradict the assumption that $f(x_N) - f_*$ is the worst-case. Hence, the worst-case value for the criteria $f(x_N) - f_*$ scales with L.

Using a similar argument with f_2, we conclude that $f(x_N) - f_*$ scales with R^2. □

4.2.6 A simple example

Consider the very simple case of a method performing a single gradient step using the non-standard step-size $\frac{3}{2L}$, i.e., $x_1 = x_0 - \frac{3}{2L}\nabla f(x_0)$ (this is actually the best possible step size for a single step for smooth convex unconstrained minimization, see Appendix 4.A for a more detailed treatment of a single iteration and see Section 4.3.1 for more iterations). One wishes to estimate the worst-case objective function accuracy after taking that step, i.e., maximize $f(x_1) - f_*$, over all L-smooth convex functions. Solving the corresponding semidefinite formulation (sdp-PEP) with $\mu = 0$, $N = 1$, $H = \left(\frac{3}{2}\right)$ and $\mathcal{P}_{b,C}(f, G) = f_1$ provides the optimal value

$$w_{0,L}^{\text{sdp}}\left(R, \left(\tfrac{3}{2}\right), 1, \begin{pmatrix}0\\1\end{pmatrix}, 0^{3\times 3}\right) = \frac{LR^2}{8},$$

attained by the following optimal solution with *rank one* Gram matrix G

$$f_0 = \frac{LR^2}{2}, f_1 = \frac{LR^2}{8} \text{ and } G = LR^2\begin{pmatrix} L & -L/2 & 1 \\ -L/2 & L/4 & -1/2 \\ 1 & -1/2 & 1/L \end{pmatrix} \succeq 0.$$

This means that $f(x_1) - f_* \leq \frac{LR^2}{8}$ holds for any $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$ for any d and provided that $\|x_0 - x_*\| \leq R$. It is easy to check that function $f(x) = \frac{L}{2}x^2 \in \mathcal{F}_{0,L}(\mathbb{R})$ achieves this worst-case when started from $x_0 = R$. Indeed one can successively evaluate $f_0 = f(x_0) = \frac{LR^2}{2}$, $g_0 = \nabla f(x_0) = LR$, $x_1 = R - \frac{3}{2}R = -\frac{R}{2}$, $f_1 = f(x_1) = \frac{LR^2}{8}$ and $g_1 = -\frac{LR}{2}$. This function is one-dimensional since the optimal G has rank one (note that Corollary 4.5 only guaranteed the existence of a three-dimensional worst-case).

Solving the dual problem (d-sdp-PEP) leads to the same optimal value $\frac{LR^2}{8}$, attained by optimal multipliers $\lambda_{01} = \lambda_{*0} = \lambda_{*1} = \frac{1}{2} \geq 0$ and $\tau = \frac{L}{8}$. The corresponding dual slack matrix is

$$S = \frac{1}{2}\begin{pmatrix} 1/L & 1/L & -1/2 \\ 1/L & 1/L & -1/2 \\ -1/2 & -1/2 & L/4 \end{pmatrix} = \frac{L}{2}\begin{pmatrix} -1/L \\ -1/L \\ 1/2 \end{pmatrix}\begin{pmatrix} -1/L & -1/L & 1/2 \end{pmatrix} \succeq 0.$$

From this dual solution, a fully explicit proof of the worst-case performance can be derived, which can be checked independently without any knowledge about our approach. Indeed, linear equalities in the dual imply that the objective accuracy $f(x_1) - f_*$ can be written exactly as follows

$$
\begin{aligned}
&f(x_1) - f(x_*) \\
&= \frac{1}{2}\left(f(x_1) - f(x_0) + \nabla f(x_1)^\top (x_0 - x_1) + \frac{1}{2L}\|\nabla f(x_0) - \nabla f(x_1)\|_2^2\right) \\
&+\frac{1}{2}\left(f(x_0) - f(x_*) + \nabla f(x_0)^\top (x_* - x_0) + \frac{1}{2L}\|\nabla f(x_0) - \nabla f(x_*)\|_2^2\right) \\
&+\frac{1}{2}\left(f(x_1) - f(x_*) + \nabla f(x_1)^\top (x_* - x_1) + \frac{1}{2L}\|\nabla f(x_1) - \nabla f(x_*)\|_2^2\right) \\
&+\frac{L}{8}\|x_0 - x_*\|^2 - \frac{L}{2}\left\|\frac{1}{2}(x_0 - x_*) - \frac{\nabla f(x_0)}{L} - \frac{\nabla f(x_1)}{L}\right\|^2
\end{aligned}
$$

(where for the last term we write the quadratic form $\mathrm{Tr}\,(SG)$ as a square, since S is rank-one). This equality, which is straightforward to check using $x_1 = x_0 - \frac{3}{2L}\nabla f(x_0)$ and $\nabla f(x_*) = 0$, immediately implies inequality $f(x_1) - f_* \leq \frac{L}{8}\|x_0 - x_*\|^2$, since the first three bracketed expressions are nonpositive because of inequalites from Corollary 3.9 valid for all functions in $\mathcal{F}_{0,L}$.

4.3 Study of standard first-order methods

In this section we apply the convex PEP formulation to study convergence of the fixed-step gradient method (GM), the standard fast gradient method (FGM) and the optimized gradient method (OGM) proposed by [KF16d].

We begin with the GM for smooth convex optimization, whose worst-case is conjectured in [DT14] to be attained on a simple one-dimensional function. Numerical experiments with our exact formulation confirm this conjecture. Further experiments on the worst-case complexity for different methods, problem classes and performance criteria lead to a series of conjectures based on worst-case functions possessing a similar shape. We conclude this section with the study of a nonlinear performance criteria corresponding to the smallest gradient norm among all iterates computed by the method.

The results were obtained using the *large-scale regime* formulation of PEP (without rank constraint), and are essentially numerical. They were obtained on an Intel 3.5Ghz desktop computer using a combination of the YALMIP modeling environment in MATLAB [LÖ4], the MOSEK [Mos10] and SeDuMi [Stu99] semidefinite solvers and the VSDP (verified semidefinite programming) toolbox [HJL12].

Remark 4.8. Note that OGM has now been extensively studied by Kim and Fessler in [KF16c, KF16b] (see survey in Section 1.3.2), by Drori in [Dro16] (see Section 1.3.2) and been extended for composite minimization (see Chapter 5 and [KF16a, THG16b]). In addition, a projected version of the gradient method

was studied in Drori's thesis [Dro14] (see Section 1.3.2), whereas its proximal version for smooth (possibly strongly) convex composite optimization is further studied in Chapter 7 (or [THG16c]).

4.3.1 Gradient method

As previously underlined, we begin by a numerical validation of a recent conjecture by Drori and Teboulle [DT14] on the behavior of the gradient method for smooth convex unconstrained minimization for the objective function accuracy $f(x_N) - f(x_*)$.

After that, we extend the conjecture to the smooth strongly convex unconstrained minimization setting in both function value accuracy $f(x_N) - f(x_*)$ and residual gradient norm $\|\nabla f(x_N)\|$. On the way, we discuss the optimal values of the step size parameter for the different settings.

For doing that, we rely on the formulation (sdp-PEP). Extensive numerical validations suggest that the corresponding SDP provides us with one-dimensional worst-case functions and hence, that they are also the solutions to sdp-PEP(d) for any $d \geq 1$.

Conjecture on smooth convex functions by Drori and Teboulle [DT14]

Consider the classical fixed-step gradient method (GM) with constant step sizes applied to a smooth convex function in $\mathcal{F}_{0,L}(\mathbb{R}^d)$. Following the discussion in section 4.2.5 we use normalized step sizes $\frac{h}{L}$, inversely proportional to L.

Gradient Method (GM)

Input: $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$, $x_0 \in \mathbb{R}^d$, $y_0 = x_0$.

For $i = 0 : N-1$

$$x_{i+1} = x_i - \frac{h}{L}\nabla f(x_i)$$

The following conjecture on the convergence of the worst-case objective function values was made in [DT14].

Conjecture 4.9 ([DT14], Conjecture 3.1.)**.** Any sequence of iterates $\{x_i\}$ generated by the gradient method GM with constant normalized step size $0 \leq h \leq 2$ on a smooth convex function $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$ satisfies

$$f(x_N) - f_* \leq \frac{LR^2}{2} \max\left(\frac{1}{2Nh+1}, (1-h)^{2N}\right).$$

A proof of the conjecture is provided in [DT14] for step sizes $0 \leq h \leq 1$, leaving the case $1 < h < 2$ open. We also recall that the upper bound in this conjecture cannot be improved, as it matches the performance of the GM on two specific one-dimensional functions. Indeed, define

$$f_1(x) = \begin{cases} \frac{LR}{2Nh+1}|x| - \frac{LR^2}{2(2Nh+1)^2} & \text{if } |x| \geq \frac{R}{2Nh+1}, \\ \frac{L}{2}x^2 & \text{else,} \end{cases}$$
$$f_2(x) = \frac{L}{2}x^2.$$

It is straightforward to check that the final objective value accuracy of GM on f_1 is equal to $\frac{LR^2}{2}\frac{1}{2Nh+1}$, and that it is equal to $\frac{LR^2}{2}(1-h)^{2N}$ on f_2. This means that the conjecture can be reformulated as saying that the worst-case behavior of the GM according to objective function accuracy is achieved by function f_1 or f_2, depending on which of the two is worst (which will depend only on the normalized step size h and number of iterations N).

Intuitively, the behavior of GM on piecewise affine-quadratic f_1 corresponds to a situation in which iterates slowly approach the optimal value without oscillating around it (i.e., no overshooting), whereas GM applied on purely quadratic f_2 generates a sequence oscillating around the optimal point. Those behaviors are illustrated on Figure 4.1. We also note that iterates for f_1 stay on the affine piece of the function, and even never come close to the quadratic piece. Interestingly, the existence of a one-dimensional worst-case function with a simple affine-quadratic shape will also be observed for the other algorithms studied in this section, both in the smooth convex and in the smooth strongly convex cases.

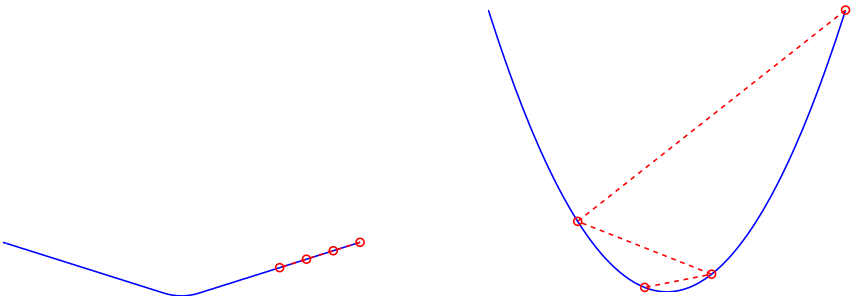


Figure 4.1: Behavior of the gradient method on f_1 (left) and f_2 (right), for $L = R = 1$. We observe that GM does not overshoot the optimal solution on f_1, while it does so at each iteration on f_2.

Empirical results from the numerical resolution of (sdp-PEP) strongly support Conjecture 4.9. Indeed, when comparing its predictions with numerically computed worst-case bounds, we obtained a maximal relative error of magni-

tude 10^{-7} (all pairs of values $N \in \{1, 2, \ldots, 30\}$ and $h \in \{0.05, 0.10, \ldots, 1.95\}$ were tested). It is also worth pointing out that the Gram matrices computed numerically correspond to the one-dimensional worst-case functions f_1 and f_2 introduced above.

Optimal step sizes. Before going into the details of other methods, we underline another observation coming from [DT14]: Conjecture 4.9 also suggests the existence of an optimal step size $h_{\text{opt}}(N)$ for the GM — optimal in the sense of achieving the lowest worst-case. That is, if you know in advance how many iterations of the GM you will perform, it suggests using a step size $h_{\text{opt}}(N)$ that is the unique minimizer of the right-hand side of the Conjecture 4.9 for a fixed value of N. It is obtained by solving[2] the following non-linear equation in h_{opt} (for which no closed-form solution seems to be available):

$$\frac{1}{2Nh_{\text{opt}} + 1} = (1 - h_{\text{opt}})^{2N}.$$

This optimal step size can be interpreted in terms of the trade-off between what we obtain on functions f_1 and f_2. On the one hand, we ensure that we are not going too slowly to the optimal point on f_1, and on the other hand we do not want to overshoot too much on f_2.

Assuming Conjecture 4.9 holds true, one can show that the optimal step size is an increasing function of N with $3/2 \leq h_{\text{opt}}(N) < 2$ and $h_{\text{opt}}(N) \to 2$ as $N \to \infty$. More precisely, working out the expression defining h_{opt} gives the following tight lower and upper estimates[3]:

$$\begin{aligned} 2 - \frac{\log 4N}{2N} &\sim 1 + (1 + 4N)^{-1/(2N)} \\ &\leq h_{\text{opt}}(N) \leq 1 + (1 + 2N)^{-1/(2N)} \sim 2 - \frac{\log 2N}{2N}. \end{aligned} \tag{4.3}$$

It is interesting to compare the results from the relaxation (G') proposed for GM in [DT14] with ours, for values of the normalized step size h that are close to h_{opt}. Indeed, while the results of the two formulations are quite similar for most values of h, it turns out that those from [DT14] are significantly more conservative in the zone around h_{opt}, as presented in Table 4.1 for different values

[2]This equation possesses several solutions, but the optimum is the unique point where the two terms feature derivatives of opposite signs (a necessary and sufficient condition for the maximum of two convex functions of one variable). This point can easily be computed numerically with an appropriate bisection method.

[3]Note that a bit of sensitivity analysis shows that it is preferable to use a lower bound on $h_{\text{opt}}(N)$ rather than an upper bound. Hence, the use of the approximate $h_{\text{opt}}(N) = 1 + (1 + 4N)^{-1/(2N)}$ should be favored over $h_{\text{opt}}(N) = 1 + (1 + 2N)^{-1/(2N)}$. Also, it is preferable to underestimate the maximum number of iterations (and therefore to use a step size smaller than h_{opt}) than to overestimate it.

of N. This also formally establishes the fact that the formulation from [DT14] is a strict relaxation of the performance estimation problem.

These numerical results have been obtained with MOSEK, a standard semidefinite optimization solver. Despite convexity of the formulation, it might happen that the solution returned by such as solver is inaccurate, and in particular (slightly) infeasible. In that case, the objective value of the approximate primal (resp. dual) solution is no longer guaranteed to be a lower (resp. upper) bound on the exact optimal value, hence potentially negating the advantage of an exact convex formulation. For this reason, all numerical results reported in this section have been double checked with an interval arithmetic-based semidefinite optimization solver [HJL12] that returns an interval that is guaranteed to contain the optimal value. These guaranteed bounds are reported in Table 4.2 for the case $h = 1.5$, which compares them with Conjecture 4.9.

We can observe that the use of a verified solver does not impact our conclusions about the validity of the conjecture. Moreover, this table is typical of what we observed for all conjectures in this section: all numerical results reported were validated[4], and in what follows we will no longer mention this verification explicitly.

Finally, we compare results obtained with Conjecture 4.9 with classical analytical bounds from the literature for the GM with unit normalized step size $h = 1$ (which is usually recommended, and sometimes called optimal). The best analytical bound we could find, e.g. in [Ber15, Proposition 6.1.7], states that

$$f(x_N) - f_* \leq \frac{LR^2}{2N}. \tag{4.4}$$

This analytical bound is asymptotically worse by a factor of 2 than the bound predicted by Conjecture 4.9 with $h = 1$. Similarly, one can investigate the effect of choosing the optimal normalized step size $h_{\text{opt}}(N)$ instead of $h = 1$: Conjecture 4.9 then predicts another improvement by a factor of 2. These observations follow from the asymptotic (large N) behaviors of the different worst-case bounds on $f(x_N) - f_*$, which can easily be computed:

$$\text{Conjecture 4.9 with } h = 1 : \frac{\max\{f(x_N) - f(x_*)\}}{\frac{LR^2}{2}\frac{1}{2N+1}} = 1,$$

$$\text{Conjecture 4.9 with } h = h_{\text{opt}}(N) \ : \lim_{N\to\infty} \frac{\max\{f(x_N) - f(x_*)\}}{\frac{LR^2}{2}\frac{1}{4N+1}} = 1.$$

[4]Except for tests where validation encountered numerical difficulties, i.e for which VSDP returned no valid interval, which occurred more and more frequently as the value of the worst-case bound became closer to zero.

N	h_{opt}	Conjecture 4.9	Value computed in [DT14]	Rel. error	Value from (sdp-PEP)	Rel. error
1	1.5000	$LR^2/8.00$	$LR^2/8.00$	0.00	$LR^2/8.00$	7e-09
2	1.6058	$LR^2/14.85$	$LR^2/14.54$	2e-02	$LR^2/14.85$	5e-09
5	1.7471	$LR^2/36.94$	$LR^2/32.57$	1e-01	$LR^2/36.94$	1e-08
10	1.8341	$LR^2/75.36$	$LR^2/59.80$	3e-01	$LR^2/75.36$	3e-08
20	1.8971	$LR^2/153.77$	$LR^2/109.58$	4e-01	$LR^2/153.77$	6e-08
30	1.9238	$LR^2/232.85$	$LR^2/156.23$	5e-01	$LR^2/232.85$	7e-08
40	1.9388	$LR^2/312.21$	$LR^2/201.10$	6e-01	$LR^2/312.21$	3e-08
50	1.9486	$LR^2/391.72$	$LR^2/244.70$	6e-01	$LR^2/391.72$	1e-07
100	1.9705	$LR^2/790.22$	$LR^2/451.72$	7e-01	$LR^2/790.22$	1e-07

Table 4.1: Gradient Method with $\mu = 0$, worst-case computed with relaxation from [DT14] and worst-case obtained by exact formulation (sdp-PEP) for the criterion $f(x_N) - f^*$. Error is measured relatively to the conjectured result. Results obtained with MOSEK [Mos10].

N	1	2	5	10	15	20	30
Relative error (upper limit)	2e-09	7e-10	2e-09	1e-09	9e-10	1e-09	9e-10
Conjecture	1.2e-01	7.1e-02	3.1e-02	1.6e-02	1.1e-02	8.2e-03	5.5e-03
Relative error (lower limit)	2e-09	3e-09	9e-09	9e-08	2e-07	3e-07	9e-07

Table 4.2: Gradient method with relative step size $h = 1.5$: numerical values from Conjecture 4.9 and relative error for the upper and lower limits of the guaranteed interval obtained numerically with VSDP [HJL12] and SeDuMi [Stu99].

A generalized conjecture for strongly convex functions

In view of the encouraging results obtained for the GM in the smooth case, we now study the behavior of the GM on the class of strongly convex functions $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$ using our formulation (sdp-PEP) with the same performance criterion, objective function accuracy. It turns out that the solution for every problem consisted again in a one-dimensional worst-case function (rank $G = 1$) of the same piecewise quadratic type. We therefore introduce the following general definitions for functions $f_{1,\tau}$ and f_2:

$$f_{1,\tau}(x) = \begin{cases} \frac{\mu}{2}x^2 + a_\tau|x| + b_\tau & \text{if } |x| \geq \tau \\ \frac{L}{2}x^2 & \text{else,} \end{cases}$$

$$f_2(x) = \frac{L}{2}x^2,$$

where scalars $a_\tau = (L-\mu)\tau$ and $b_\tau = -\left(\frac{L-\mu}{2}\right)\tau^2$ are chosen to ensure continuity of $f_{1,\tau}$ and its first derivative, and τ is a parameter that controls the radius of the central quadratic piece (with the largest curvature). Although the value of parameter τ could in principle be estimated from the numerical solutions of our problems, it turns out it can be computed analytically by maximizing the final objective value $f_{1,\tau}(x_N)$ (assuming that all iterates stay in the affine zone $|x| \geq \tau$), which then leads to

$$\tau = \frac{R\kappa}{(\kappa-1) + (1-\kappa h)^{-2N}} \tag{4.5}$$

where $\kappa = \frac{\mu}{L}$ is the inverse condition number of the problem class $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$. We are now able to extend Conjecture 4.9 to the GM applied to strongly convex functions.

Conjecture 4.10. Any sequence of iterates $\{x_i\}$ generated by the gradient method GM with constant normalized step sizes $0 \leq h \leq 2$ on a smooth strongly convex function $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ satisfies

$$f(x_N) - f_* \leq \frac{LR^2}{2} \max\left(\frac{\kappa}{(\kappa-1) + (1-\kappa h)^{-2N}}, (1-h)^{2N}\right).$$

As in the previous section, this conjecture states that the worst-case behavior of the GM according to objective function accuracy is achieved by function $f_{1,\tau}$ or f_2, depending on which of the two is worse. Proceeding now to numerical validations, we first point out that our results are intrinsically limited to the accuracy that can be reached by numerical SDP solvers. For this reason, we only report on situations for which Conjecture 4.10 predicts a final accuracy larger than 10^{-6}, ensuring a few significant digits for the numerical results.

The resulting estimated relative differences between Conjecture 4.10 and the numerical results obtained with (sdp-PEP) are given in Table 4.3, for different values of κ. We observe that the conjecture is very well supported by our numerical results, with a largest relative error around 10^{-6}, reached for the largest value of κ considered here. This is expected as GM tends to perform better as κ increases (i.e., final accuracy $f(x_N) - f_*$ approaches zero), which renders a precise comparison between numerical results and the conjecture more and more difficult.

κ	0	.001	.005	.010	.015	0.1	0.2	0.5
Relative error	6e-10	7e-10	4e-10	6e-10	8e-10	2e-07	9e-08	1e-06

Table 4.3: Maximum relative estimated differences between Conjecture 4.10 and corresponding numerical results obtained with SeDuMi [Stu99]. The maximum is taken over all $N \in \{1, \dots, 30\}$ and $h \in \{0.05, \dots, 1.95\}$ for which the conjecture predicts a worst-case larger than 10^{-6}.

We now investigate some consequences of our conjecture. First, we note that Conjecture 4.10 tends to Conjecture 4.9 as μ tends to zero. This is a consequence of the fact that τ tends to $\frac{R}{2Nh+1}$ as κ tends to zero (one can also check that function $f_{1,\tau}$ tends to function f_1 introduced earlier). Hence our formulation (sdp-PEP) closes an apparent gap between worst-case analyses of the smooth convex and the smooth strongly convex cases. Indeed, to the best of our knowledge, existing worst-case bounds for the smooth strongly convex case do not converge to the smooth case as $\mu \to 0$.

It is also interesting to compare our results to those obtained with the IQC methodology of [LRP16]. If we only care about asymptotic linear rates of convergence, Conjecture 4.10 predicts

$$f(x_N) - f_* \leq \frac{LR^2}{2} \max\{\kappa\, \rho_1^{2N}, \rho_2^{2N}\} \text{ with } \rho_1 = |1 - \kappa h| \text{ and } \rho_2 = |1 - h|$$

(the first term in the max was obtained by neglecting $(\kappa - 1)$ in the denominator). On the other hand [LRP16, Section 4.4] proves that the distance to the solution converges linearly according to

$$\|x_N - x_*\| \leq \rho^N \|x_0 - x_*\| \text{ with a factor } \rho = \max\{\rho_1, \rho_2\}$$

with the same values for ρ_1 and ρ_2. This matches our asymptotic rate up to a multiplicative constant.

Optimal step sizes. As for Conjecture 4.9, our new Conjecture 4.10 suggests optimal step sizes $h_{\text{opt}}(N, \kappa)$, which can be obtained by solving the equation

(for $0 < \kappa < 1$)

$$\frac{\kappa}{(\kappa-1)+(1-\kappa h_{\text{opt}})^{-2N}} = (1-h_{\text{opt}})^{2N} \tag{4.6}$$

(note that one recovers the previous equation for h_{opt} when μ tends to zero). For a given N, as κ increases from 0 to 1, those optimal step sizes decrease from $h_{\text{opt}}(N,0)$ (optimal step size in the smooth case) to $h_{\text{opt}}(N,1) = 1$ (the latter being expected since it can only correspond to the case of function f_2 in the original (PEP), for which the GM with $h = 1$ converges in one iteration). For a given κ, we find that $h_{\text{opt}}(N,\kappa)$ increases as N increases, as in the smooth convex case, according to the following lower and upper estimates

$$\begin{aligned} &1+\left(\frac{\kappa-1}{\kappa}+\frac{1}{\kappa}\Big(\frac{1+\kappa}{1-\kappa}\Big)^{2N}\right)^{-\frac{1}{2N}} \\ &\quad\leq h_{\text{opt}}(N,\kappa) \leq \min\left\{1+\left(\frac{(\kappa-1)}{\kappa}+\frac{1}{\kappa}(1-\kappa)^{-2N}\right)^{-\frac{1}{2N}}, \frac{2}{1+\kappa}\right\} \end{aligned} \tag{4.7}$$

which both tend to $\frac{2}{1+\kappa}$ as N increases (the first term appearing in the min of the upper bound tends to $2-\kappa$, which is always greater than $\frac{2}{1+\kappa}$). This limiting normalized step size $\frac{2}{1+\kappa}$ corresponds to step size $\frac{2}{L+\mu}$ that is often recommended for the GM, and sometimes called optimal.

We now illustrate the improvements provided by Conjecture 4.10 with respect to the classical analytical worst-case bound found in the literature. When using normalized step size $h = \frac{2}{1+\kappa}$, iterates from GM applied to functions in $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$ are known to satisfy (see [Nes04, Theorem 2.1.14] for example)

$$f(x_N) - f_* \leq \frac{LR^2}{2}\left(\frac{1-\kappa}{1+\kappa}\right)^{2N}. \tag{4.8}$$

On the other hand, as the number of steps N tends to infinity, the true worst-case predicted by Conjecture 4.10 for the same step size asymptotically tends to $\frac{LR^2}{2}\big(\frac{1-\kappa}{1+\kappa}\big)^{2N}$, which is exactly the same as (4.8). Indeed, one can check that this rate is equal to the second term appearing in the max of Conjecture 4.10, while the first term tends to $\frac{LR^2}{2}\kappa\big(\frac{1-\kappa}{1+\kappa}\big)^{2N}$ which is always smaller.

One can however do better using the optimal step size h_{opt}. Since it is not closed-form, we use the following approximate expression obtained after solving a suitable approximation of equation (4.6)

$$\tilde{h}_{\text{opt}}(N) = \frac{1+\kappa^{\frac{1}{2N}}}{1+\kappa^{1+\frac{1}{2N}}}$$

(note that $\tilde{h}_{\text{opt}}(N)$ tends to $\frac{2}{1+\kappa}$ as N grows), and find that Conjecture 4.10

predicts a worst-case with asymptotic convergence rate $\left(\frac{1-\kappa}{1+\kappa}\right)^2$:

$$\lim_{N\to\infty} \frac{\max\{f(x_N) - f(x_*)\}}{\frac{LR^2}{2}\left(\frac{1-\kappa}{1+\kappa}\right)^{2N}} = \kappa^{\frac{1}{1+\kappa}}$$

which improves the asymptotic rate by a factor $\left(\frac{1}{\kappa}\right)^{\frac{1}{1+\kappa}}$ (which can be shown to lie between $\frac{3}{4}\frac{1}{\kappa}$ and $\frac{1}{\kappa}$).

A conjecture on the gradient norm

We now consider a different performance criterion, given by the norm of the gradient computed at the last iterate. Numerical experiments with our formulation suggest that results similar to those presented in the previous sections can be obtained both in the smooth convex and smooth strongly convex cases, based again on one-dimensional piecewise quadratic worst-case functions. Using the same definition for functions $f_{1,\tau}$ and f_2, and choosing now the parameter τ according to

$$\tau = \frac{R\kappa}{(\kappa-1) + (1-\kappa h)^{-N}}, \tag{4.9}$$

we propose the following conjecture.

Conjecture 4.11. Any sequence of iterates $\{x_i\}$ generated by the gradient method GM with constant normalized step sizes $0 \leq h \leq 2$ on a smooth strongly convex function $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ satisfies

$$\|\nabla f(x_N)\|_2 \leq LR \max\left(\frac{\kappa}{(\kappa-1) + (1-\kappa h)^{-N}}, |1-h|^N\right).$$

As for Conjecture 4.10, we limit our numerical validation to the cases where the worst-case values predicted by the Conjecture are larger than 10^{-6}; the largest relative error is about 10^{-7}.

We note that, as κ tends to zero (i.e., the smooth case), Conjecture 4.11 tends to

$$\|\nabla f(x_N)\|_2 \leq LR \max\left(\frac{1}{Nh+1}, |1-h|^N\right).$$

Optimal step sizes. From that, we see that the optimal step size $h^{\nabla}_{\text{opt}}(N,0)$ for the GM is again an increasing function of N with $\sqrt{2} \leq h^{\nabla}_{\text{opt}}(N,0) < 2$ and $h^{\nabla}_{\text{opt}}(N,0) \to 2$ as $N \to \infty$. In the strongly convex case $\kappa > 0$, the optimal step size is a decreasing function of κ and satisfies $h^{\nabla}_{\text{opt}}(N,\kappa) \to 1$ as $\kappa \to 1$. As in

the previous case, $h^{\nabla}_{\text{opt}}(N,\kappa)$ is bounded above by $\frac{2}{1+\kappa}$, which we can confirm with the following lower and upper bounds on h^{∇}_{opt}:

$$1+\left(\frac{\kappa-1}{\kappa}+\left(\frac{1+\kappa}{1-\kappa}\right)^N\right)^{-1/N} \leq h^{\nabla}_{\text{opt}}(N,\kappa) \leq \min\left\{1+\left(\frac{\kappa-1}{\kappa}+\frac{1}{\kappa}(1-\kappa)^{-N}\right)^{-1/N}, \frac{2}{1+\kappa}\right\}.$$

In the smooth case, those bounds reduce to the simpler expression

$$2-\frac{\log 2N}{N} \sim 1+(1+2N)^{-1/N} \leq h^{\nabla}_{\text{opt}} \leq 1+(1+N)^{-1/N} \sim 2-\frac{\log N}{N}.$$

We now compare with a standard analytical worst-case bound. The iterates of the GM method with normalized step size $\frac{2}{1+\kappa}$ are known to satisfy

$$\begin{aligned} &\|x_N - x_*\|_2 \leq R\left(\frac{1-\kappa}{1+\kappa}\right)^N \text{ and} \\ &\|\nabla f(x_N)\|_2 \leq L\|x_N - x_*\|_2 \leq LR\left(\frac{1-\kappa}{1+\kappa}\right)^N \end{aligned} \tag{4.10}$$

(see for example [Nes04] for the left inequality, and use the L-Lipschitz property of the gradient along with $\nabla f(x_*)=0$ to derive the right inequality). The latter estimate is tight according to Conjecture 4.11. Using the following approximate optimal step size

$$\tilde{h}^{\nabla}_{\text{opt}}(N) = \frac{1+\kappa^{\frac{1}{N}}}{1+\kappa^{1+\frac{1}{N}}}$$

(which tends to $\frac{2}{1+\kappa}$ as N grows) can be shown to improve the conjectured asymptotic rate by the same factor $\kappa^{-\frac{1}{1+\kappa}}$ as for convergence in function values.

4.3.2 Fast gradient and optimized gradient methods

In this section we assess the performance in the smooth convex case ($\mu = 0$) of two accelerated first-order methods: the so-called fast gradient method (FGM) due to Nesterov [Nes83], and an optimized gradient method (OGM) recently proposed by Kim and Fessler [KF16d].

Fast Gradient Method (FGM)

Input: $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$, $x_0 \in \mathbb{R}^d$, $y_0 = x_0$, $\theta_0 = 1$.

For $i = 0 : N-1$

$$\begin{aligned} y_{i+1} &= x_i - \frac{1}{L}\nabla f(x_i) \\ \theta_{i+1} &= \frac{1+\sqrt{4\theta_i^2+1}}{2} \\ x_{i+1} &= y_{i+1} + \frac{\theta_i - 1}{\theta_{i+1}}(y_{i+1} - y_i) \end{aligned}$$

Optimized Gradient Method (OGM)

Input: $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$, $x_0 \in \mathbb{R}^d$, $y_0 = x_0$, $\theta_0 = 1$.

For $i = 0 : N-1$

$$\begin{aligned} y_{i+1} &= x_i - \frac{1}{L}\nabla f(x_i) \\ \theta_{i+1} &= \begin{cases} \frac{1+\sqrt{4\theta_i^2+1}}{2}, & i \leq N-2 \\ \frac{1+\sqrt{8\theta_i^2+1}}{2}, & i = N-1 \end{cases} \\ x_{i+1} &= y_{i+1} + \frac{\theta_i - 1}{\theta_{i+1}}(y_{i+1} - y_i) + \frac{\theta_i}{\theta_{i+1}}(y_{i+1} - x_i) \end{aligned}$$

Both of these algorithms are defined in terms of two sequences: $\{y_i\}_i$ is a primary sequence, and $\{x_i\}_i$ is a secondary sequence, where the gradient is evaluated. We first show that both of these algorithms can be expressed as fixed-step first-order methods, which we defined as

$$x_i = x_0 - \sum_{k=0}^{i-1} h_{i,k}\nabla f(x_k) \quad (\text{for } L = 1).$$

One way to proceed is to focus on the secondary sequence $\{x_i\}_i$ and substitute the y_i's in the algorithm formulation. For FGM, we have

$$\begin{aligned} x_{i+1} &= x_i - \frac{g_i}{L} + \frac{\theta_i - 1}{\theta_{i+1}}\left(x_i - x_{i-1} - \frac{g_i}{L} + \frac{g_{i-1}}{L}\right), \\ &= x_i + \frac{\theta_i - 1}{\theta_{i+1}}(x_i - x_{i-1}) - \left(\frac{\theta_i - 1}{\theta_{i+1}} + 1\right)\frac{g_i}{L} + \frac{\theta_i - 1}{\theta_{i+1}}\frac{g_{i-1}}{L}, \end{aligned}$$

which allows to obtain the step sizes relative to x_0 by recurrence:

$$h_{i+1,k} = \begin{cases} h_{i,k} + \frac{\theta_i - 1}{\theta_{i+1}} (h_{i,k} - h_{i-1,k}) & \text{if } k \leq i-2, \\ h_{i,k} + \frac{\theta_i - 1}{\theta_{i+1}} (h_{i,k} - 1) & \text{if } k = i-1, \\ \frac{\theta_i - 1}{\theta_{i+1}} + 1 & \text{if } k = i, \end{cases}$$

with initial conditions $h_{1,0} = 1$, $h_{1,k} = 0$ if $k < 0$ and $h_{0,k} = 0$ for all k. Similarly, we have for OGM

$$h_{i+1,k} = \begin{cases} h_{i,k} + \frac{\theta_i - 1}{\theta_{i+1}} (h_{i,k} - h_{i-1,k}) & \text{if } k \leq i-2, \\ h_{i,k} + \frac{\theta_i - 1}{\theta_{i+1}} (h_{i,k} - 1) & \text{if } k = i-1, \\ \frac{2\theta_i - 1}{\theta_{i+1}} + 1 & \text{if } k = i, \end{cases}$$

with the same initial conditions. This approach will provide estimates for the last secondary iterate x_N. If an estimate for last primary iterate y_N is needed, one just has to replace the expression of x_N by y_N, which is done by using the following alternative coefficients for the last step:

$$h_{N,k} = \begin{cases} h_{N-1,k} & \text{if } k \leq N-2, \\ 1 & \text{if } k = N-1, \end{cases}$$

for both FGM and OGM.

Again, our numerical experiments strongly suggest the same assumption about the shape of the worst-case functions, i.e., one-dimensional and piecewise quadratic (with iterates staying in the affine zone of $f_{1,\tau}$). Using this property, we are able to compute the following values of τ achieving the worst-case final objective accuracy, which surprisingly hold for both the classical FGM and the more recent OGM (a coincidence for which we can offer no explanation)

$$\tau_1 = \frac{R}{2\sum_{k=0}^{N-2} h_{N-1,k} + 3} \text{ for the primary sequence,}$$
$$\tau_2 = \frac{R}{2\sum_{k=0}^{N-1} h_{N,k} + 1} \text{ for the secondary sequence.}$$

Our numerical results suggest the following two conjectures (validations for both conjectures were performed for values of $N \in \{1, \ldots, 100\}$ and displayed a relative error less than 10^{-4}).

Conjecture 4.12. Any (primary) sequence of iterates $\{y_i\}$ generated by the fast gradient methd FGM (resp. optimized gradient method OGM) on a smooth convex function $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$ satisfies

$$f(y_N) - f_* \leq f_{1,\tau_1}(y_{1,N}) = \frac{LR^2}{2} \frac{1}{2\sum_{k=0}^{N-2} h_{N-1,k} + 3},$$

where $y_{1,N}$ is the final (primary) iterate computed by FGM (resp. OGM) applied to f_{1,τ_1} starting from $x_0 = R$, and quantities $h_{N-1,k}$ are the fixed coefficients of the last step of FGM (resp. OGM).

Conjecture 4.13. Any (secondary) sequence of iterates $\{x_i\}$ generated by the fast gradient methd FGM (resp. optimized gradient method OGM) on a smooth convex function $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$ satisfies

$$f(x_N) - f_* \leq f_{1,\tau_2}(x_{1,N}) = \frac{LR^2}{2} \frac{1}{2\sum_{k=0}^{N-1} h_{N,k} + 1},$$

where $x_{1,N}$ is the final (secondary) iterate computed by FGM (resp. OGM) applied to f_{1,τ_2} starting from $x_0 = R$, and quantities $h_{N,k}$ are the fixed coefficients of the last step of FGM (resp. OGM).

Note that Conjecture 4.13 has now been proved by Kim and Fessler [KF16d] (revised version) in the case of OGM.

The worst-case bounds in these two conjectures involve the normalized step sizes of the FGM and OGM methods. It turns out these can be computed in closed form for OGM (see also [KF16c]), and give ($N \geq 1$)

$$f(y_N) - f_* \leq \frac{LR^2}{4\theta_{N-1}^2 + 2} \leq \frac{LR^2}{2} \frac{2}{(N+1)^2 + 2}, \quad \text{and}$$
$$f(x_N) - f_* \leq \frac{LR^2}{2\theta_N^2} \leq \frac{LR^2}{2} \frac{2}{(N+1)(N+1+\sqrt{2})}$$

(where the inequalities rely on $\theta_{N-1}^2 \geq \frac{(N+1)^2}{4}$ and $\theta_N^2 \geq \frac{(N+1)(N+1+\sqrt{2})}{2}$). We were not able to obtain similar closed-form bounds for the FGM.

We now compare the numerical values obtained with Conjectures 4.12 and 4.13 with analytical bounds known for the FGM. We use for the primary sequence

$$f(y_N) - f_* \leq \frac{2LR^2}{(N+1)^2}, \tag{4.11}$$

which can be found in [BT09b, Theorem 4.4], and for the secondary sequence

$$f(x_N) - f_* \leq \frac{2LR^2}{(N+2)^2} \tag{4.12}$$

which was very recently derived in [KF16d, Theorem 1]. The comparison is displayed on Figure 4.2. The asymptotic behaviors of both sequences are well captured by the analytical bounds (4.11) and (4.12), but we observe that the estimation of the transient worst cases are improved by our conjectures: a factor approximately equal to 1.15 is gained for both sequences after 30 iterations.

Before going into the next section, we comment on the applicability of our results to monotone variants of first-order methods, i.e. methods which guarantee $f(y_{i+1}) \leq f(y_i)$. Consider for example FISTA [BT09b], which is equivalent to FGM when applied to smooth unconstrained minimization. MFISTA [BT09a], a monotone variant of FISTA. As FISTA happens to generate a monotonically decreasing sequence $\{f(y_i)\}_i$ when applied to our worst-case function f_{1,τ_1} from $x_0 = R$, the corresponding lower bound from Conjecture 4 also applies to MFISTA.

4.3.3 Estimation of the smallest gradient norm among all iterates

First-order methods are often used in dual approaches where, in addition to objective function accuracy, gradient norm plays an important role. Indeed, this quantity controls primal feasibility of the iterates (see e.g., [DGN12]). Considering for example the accelerated FGM in the smooth case, we know from the previous section that the classical analytical bound on the worst-case accuracy for a function in $\mathcal{F}_{0,L}(\mathbb{R}^d)$ is given by $\frac{2LR^2}{(N+1)^2}$. From that bound, it is easy to obtain a similar bound on the last gradient norm, using Corollary 3.9 (convexity and smoothness):

$$\|\nabla f(y_N)\|_2 \leq \sqrt{2L(f(y_N) - f_*)} \leq \frac{2LR}{N+1}. \tag{4.13}$$

Observe that this asymptotic rate is significantly worse than that of the objective function accuracy, and not better than that of the gradient method GM (see Conjecture 4.11).

However, it is well-known that the norm of the gradient is not decreasing monotonically among iterates of the FGM. Hence, in this section, we will estimate the worst-case performance of FGM according to the smallest observed gradient norm among all iterates:

$$\min_{i \in \{0,\dots,N\}} \|\nabla f(y_i)\|_2.$$

In order to do so, only a slightly modified version of (sdp-PEP) is needed: this min-type objective function is representable using a new variable t for the objective and $N+1$ additional linear inequalities $t \leq \|\nabla f(y_i)\|_2^2 \Leftrightarrow t \leq G_{i,i}$ for all $0 \leq i \leq N$. Note that the maximum is still attained since this concave piecewise linear objective function is continuous.

This criterion was suggested in [Nes12b], which proposes a variant of FGM that consists in performing $N/2$ steps of the standard FGM followed by $N/2$ steps of the GM with $h = 1$. It is then theoretically established that this variant of

FGM, which we denote by MFGM, satisfies

$$\min_{i\in\{0,\dots,N\}} \|\nabla f(y_i)\|_2 \leq \frac{8LR}{N^{3/2}}, \tag{4.14}$$

an improvement compared to the rate of convergence of the gradient of the last iterate.

We now compare FGM with this modified variant MFGM using our performance estimation formulation. Figure 4.3 compares the behaviors of those methods in both their last (for FGM) and best iterates, as well as the above analytic bounds (4.13) and (4.14).

This experiment confirms that the gradient norm of the last iterate of FGM decreases according to the slower $\mathcal{O}(N^{-1})$ rate of (4.13). We also observe that both the MFGM and the original FGM achieve the same $\mathcal{O}(N^{-3/2})$ convergence rate for the smallest gradient norm, which was not known before for FGM. In addition, numerical results reported in Table 4.4 suggest that FGM performs slightly better than MFGM. Note that the convergence rate $\mathcal{O}(N^{-3/2})$ of FGM has now been proved analytically by Kim and Fessler in [KF16b], using clever relaxations of the performance estimation problem.

A regularization technique is also described in [Nes12b], featuring a $\mathcal{O}(N^{-2})$ convergence rate up to a logarithmic factor. A drawback of this approach is that it requires a bound on the distance to the optimal solution, and that the coefficients of the method explicitly depend on this bound. No fixed-step method achieving the same $\mathcal{O}(N^{-2})$ seems to be known.

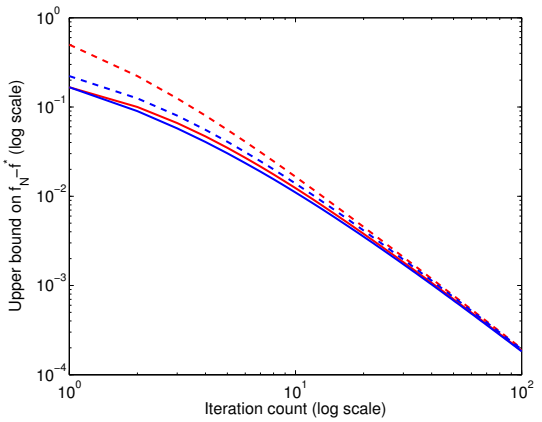


Figure 4.2: Comparison of the worst-case performance of the FGM: analytical bound (4.11) (dashed red) versus Conjecture 4.12 (red) and analytical bound (4.12) (dashed blue) versus Conjecture 4.13 (blue).

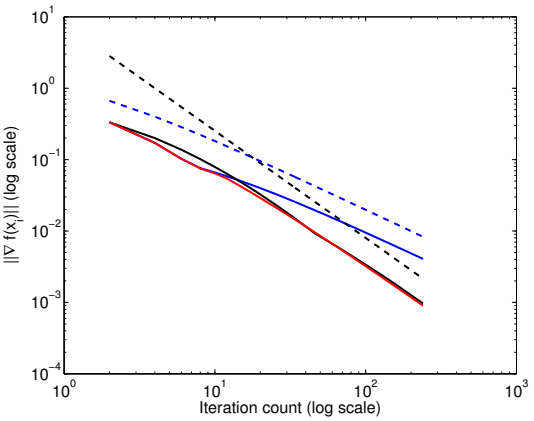


Figure 4.3: Comparison of gradient norm convergence rates for the FGM and the MFGM from [Nes12b]. Theoretical guarantees are dashed. Analytical bound on FGM (4.13) in its last iterate (dashed blue); numerical worst-case for FGM at its last iterate (blue); numerical worst-case for FGM at its best iterate (red); analytical bound on MFGM (4.14) for the best iterate (dashed black); numerical worst-case for MFGM at its best iterate (black).

N	FGM, analytic (4.13)	FGM, last	FGM, best	MFGM, analytic (4.14)	MFGM, best
2	$LR/1.50$	$LR/3.00$	$LR/3.00$	$LR/0.35$	$LR/3.00$
4	$LR/2.50$	$LR/5.84$	$LR/5.84$	$LR/1.00$	$LR/5.00$
10	$LR/5.50$	$LR/15.14$	$LR/15.62$	$LR/3.95$	$LR/12.66$
20	$LR/10.50$	$LR/25.08$	$LR/34.49$	$LR/11.18$	$LR/30.77$
30	$LR/15.50$	$LR/35.13$	$LR/58.50$	$LR/20.54$	$LR/55.38$
40	$LR/20.50$	$LR/45.19$	$LR/86.17$	$LR/31.62$	$LR/86.41$
50	$LR/25.50$	$LR/55.25$	$LR/117.08$	$LR/44.19$	$LR/119.63$
100	$LR/50.50$	$LR/105.49$	$LR/311.34$	$LR/125.00$	$LR/296.58$
200	$LR/100.50$	$LR/205.77$	$LR/850.59$	$LR/353.55$	$LR/791.87$

Table 4.4: FGM and MFGM: comparison between theoretical bounds and numerical results for the criteria $\|\nabla f(x_N)\|_2$(last) and $\min_i \|\nabla f(x_i)\|_2$ (best). Results obtained with [Mos10].

4.4 Conclusion

The contribution of this chapter is threefold: first, we presented why it was crucial to develop necessary and sufficient conditions for smooth strongly convex interpolation. Those conditions were derived by showing an explicit way of constructing the interpolating functions in Chapter 3. Second, we show that the exact worst-case performance of any fixed-step first-order algorithm for smooth strongly convex unconstrained optimization can be formulated as a convex problem. In this context, our interpolation procedure also provides explicit functions achieving the worst-case bounds computed by our approach. Third, we test of our formulation numerically on a variety of functions classes, first-order methods and performance criteria, establishing on the way a series of conjectures on the corresponding worst-case behaviors. In particular, we suggest new tight estimates of the optimal step size for the fixed-step gradient method with constant step size, which depend on the number of iterations and the condition number.

Our performance estimation problem provide a generic tool to analyze fixed-step first-order methods. It allows computing both exact worst-case guarantees and functions reaching them, and provides a unified algorithmic analysis for smooth convex functions and smooth strongly convex functions.

The exact worst-case values provided by our approach require solving a convex semidefinite program whose size grows as the square of the number of iterations considered, which may become prohibitive when this number of iterations is large. This can be avoided using iteration-independent bounds, as proposed in [LRP16], but at the cost of obtaining poorer worst-case guarantees.

Further extensions of the results and the methods presented in this chapter have already been published. Among others, the gradient method with exact line search has now been studied using PEP (see Chapter 6 or [dKGT16]). Also, Drori partially studied the projected GM in his thesis [Dro14], and the framework has been adapted to cope with composite objective functions (see Chapter 5 or [THG16b]). Also, a cyclic coordinate descent for unconstrained minimization was studied in [SL16] using a relaxed performance estimation approach. More extensions of the PEP framework can be found in Chapter 8.

In addition, various results related to optimized methods have appeared. First, as we already emphasized, the optimized gradient method for smooth unconstrained convex optimization has now been proved to be optimal by Drori in [Dro16]. Also, different extensions to OGM appeared: Kim and Fessler have further studied optimized methods for other convergence measures in [KF16b], whereas they propose a proximal extension in [KF16a].

Appendix

4.A Tight worst-case of a gradient step

Theorem 4.14. Any iterate x_1 generated by the gradient method (GM) with constant normalized step size $0 \leq h \leq 2$ on a smooth strongly convex function $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ satisfies

$$f(x_1) - f_* \leq \frac{L||x_0 - x_*||^2}{2} \max\left(\frac{\kappa}{(\kappa-1)+(1-\kappa h)^{-2}}, (1-h)^2\right),$$

with $\kappa = \frac{\mu}{L}$ the inverse condition number.

Note that this implies the following value for $h_{\text{opt}}(1) = \frac{\kappa+1-\sqrt{\kappa^2-\kappa+1}}{\kappa}$ (which tends to $\frac{3}{2L}$ as $\mu \to 0$).

Proof. The theorem can be deduced by combining the following (interpolation) inequalities with the appropriate coefficients λ_0, λ_1 and λ_2

$$\begin{aligned}
0: \quad & f_0 \geq f_1 + g_1^\top(x_0 - x_1) + \frac{1}{2L}\|g_1 - g_0\|_2^2 \\
& \qquad + \frac{\mu}{2(1-\mu/L)}\left\|x_1 - x_0 - \frac{1}{L}(g_1 - g_0)\right\|_2^2, \\
1: \quad & f_* \geq f_0 + g_0^\top(x_* - x_0) + \frac{1}{2L}\|g_0\|_2^2 \\
& \qquad + \frac{\mu}{2(1-\mu/L)}\left\|x_0 - x_* - \frac{1}{L}g_0\right\|_2^2, \\
2: \quad & f_* \geq f_1 + g_1^\top(x_* - x_1) + \frac{1}{2L}\|g_1\|_2^2 \\
& \qquad + \frac{\mu}{2(1-\mu/L)}\left\|x_1 - x_* - \frac{1}{L}g_1\right\|_2^2.
\end{aligned}$$

along with $x_1 = x_0 - \gamma g_0$ (and $\gamma = \frac{h}{L}$ the step size). Indeed, using the

coefficients (we term this solution as the *small step size regime* in the sequel)

$$\lambda_0 = \lambda_1 = \frac{1-\gamma\mu}{2-\gamma\mu}, \quad \lambda_2 = \frac{1}{2-\gamma\mu},$$

or (we call this solution the *large step size regime* in the sequel)

$$\lambda_0 = \lambda_1 = \gamma L - 1, \quad \lambda_2 = 2 - \gamma L,$$

respectively allows to obtain

$$\begin{aligned} f_1 - f_* \leq & \frac{L\|x_0 - x_*\|_2^2}{2} \frac{\kappa}{(1-\gamma\mu)^{-2} + (\kappa - 1)} \\ & - \frac{L^2}{2(L-\mu)p_1(\gamma)} \left\| x_0 - x_* + \frac{p_1(\gamma)}{(\gamma\mu - 2)} \left(\frac{g_0}{L} + \frac{g_1}{L} \right) \right\|_2^2 \\ & - \frac{p_2(\gamma)}{2(L-\mu)(2-\gamma\mu)^2} \|(\gamma\mu - 1)g_0 + g_1\|_2^2, \end{aligned}$$

or

$$\begin{aligned} f_1 - f_* \leq & \frac{L\|x_0 - x_*\|_2^2}{2} (1-\gamma L)^2 \\ & - \frac{L^2 p_3(\gamma)}{2(L-\mu)} \left\| x_0 - x_* + \frac{1 + \gamma^2 L\mu - \gamma(L+2\mu)}{L p_3(\gamma)} g_0 + \frac{\gamma L - 2}{L p_3(\gamma)} g_1 \right\|_2^2 \\ & - \frac{-p_2(\gamma)}{2(L-\mu)p_3(\gamma)} \|(\gamma L - 1)g_0 + g_1\|_2^2, \end{aligned}$$

with $p_1(\gamma) = 1 + 2\gamma(L-\mu) - \gamma^2\mu(L-\mu)$ (which is nonnegative on the interval $0 \leq \gamma \leq \frac{2}{L}$, as a simple analysis shows that the the interval $[-\mu, \frac{2}{\mu}]$ lies between the roots, and that this interval contains both 0 and $\frac{2}{L}$), $p_2(\gamma) = 3 + \gamma^2\mu L - 2\gamma(L+\mu)$ and $p_3(\gamma) = 1 - 2\gamma(L-\mu) + \gamma^2 L(L-\mu)$ (which is positive everywhere as $L \geq \mu$).

In order to conclude, we have to verify that:

(a) there is always one of the two solutions that is valid $\forall\gamma$: $0 \leq \gamma \leq \frac{2}{L}$,

(b) the valid solution is the one producing the worst value (i.e. maximum value).

We start by treating (a); in order to determine which solution is valid for a particular value of γ, let us denote by $\gamma_{\min}$ and $\gamma_{\max}$ the roots of $p_2(\gamma)$ (previously defined positive definite quadratic function). They respectively take the values

$$\gamma_{\min} = \frac{L + \mu - \sqrt{L^2 - L\mu + \mu^2}}{L\mu}, \quad \gamma_{\max} = \frac{L + \mu + \sqrt{L^2 - L\mu + \mu^2}}{L\mu}.$$

One can easily obtain that $\frac{1}{L} \leq \gamma_{\min} \leq \frac{1}{\mu}$ and $\frac{2}{L} \leq \gamma_{\max}$.

Concerning the region of validity of the two regimes, we have

- ◇ the small step size regime is valid when both $\gamma \leq \frac{1}{\mu}$ (multipliers should be positive) and $p_2(\gamma) \geq 0$ (coefficients of the norms should be positive), so γ should be outside of the interval $[\gamma_{\min},\ \gamma_{\max}]$. This regime is therefore valid for any γ such that $0 \leq \gamma \leq \gamma_{\min} \leq \frac{1}{\mu}$.
- ◇ The large step size regime is valid when both $\gamma \geq \frac{1}{L}$ (multipliers should be positive) and $p_2(\gamma) \leq 0$ (coefficients of the norms should be positive), so γ should be inside the interval $[\gamma_{\min},\ \gamma_{\max}]$. This regime is therefore valid for any γ such that $\frac{1}{L} \leq \gamma_{\min} \leq \gamma \leq \frac{2}{L} \leq \gamma_{\max}$.

Therefore, the small step size regime is valid $\forall \gamma$ such that $0 \leq \gamma \leq \gamma_{\min}$ and the large step size regime is valid in the complementary region, when $\gamma_{\min} \leq \gamma \leq \frac{2}{L}$.

In order to check which solution dominates the other, we distinguish three cases:

- ◇ in the case $0 \leq \gamma \leq \frac{1}{L}$, we have

$$\frac{\kappa}{(\kappa-1)+(1-\mu\gamma)^{-2}} \geq (1-\mu\gamma)^2 \geq (1-\gamma L)^2,$$

  so the small step size regime is indeed the one producing the larger value (in addition to being the only valid one).
- ◇ When $\frac{1}{L} \leq \gamma \leq \frac{1}{\mu}$, there can be only one intersection between the two regimes (that is, only one value of γ such that $\frac{\kappa}{(\kappa-1)+(1-\mu\gamma)^{-2}} = (1-L\gamma)^2$), since the first is a decreasing function of γ, the second is an increasing function of γ and the first is larger than the second at $\gamma = \frac{1}{L}$. Also note that $\gamma_{\min}$ is exactly the value of the intersection.
- ◇ When $\frac{1}{\mu} \leq \gamma \leq \frac{2}{L}$ (only possible when $\frac{\mu}{L} \geq \frac{1}{2}$), only the large step regime is valid. It dominates the small step regime as the value of the small step size regime is smaller at $\gamma = \frac{1}{\mu}$ and as $\frac{d}{d\mu}\left(\frac{\kappa}{(\kappa-1)+(1-\kappa h)^{-2}}\right) \geq 0$ when $\gamma \geq \frac{1}{\mu}$, with $\frac{\kappa}{(\kappa-1)+(1-\mu\gamma)^{-2}} \to (1-L\gamma)^2$ as $\kappa \to 1$.

□

Chapter 5

Performance Estimation Problems for Composite Convex Optimization

The main contributions of the chapter are the following.

- ◇ We further study the performance estimation framework, and extend it with the possibility of handling a large class of algorithms, function classes, convergence measures and initial conditions. As in Chapter 4, the approach allows formulating the worst-case estimation problem as a convex SDP using the convex interpolation framework (see Chapter 3).
- ◇ We apply the method to standard first-order methods, namely the proximal point algorithm (PPA), fixed-step projected subgradient method (PSM), different variants of fast proximal gradient methods (FPGM), a conditional gradient method (CGM) and to two alternate projection methods (APM).

Concerning the use of notations for primal and dual spaces, norms and scalar products, we refer to Section 2.1.

This chapter is divided into three main parts.

- ◇ First, Section 5.1 introduces the composite optimization problems for which the performance estimation framework of this chapter is tailored, and reviews the necessary concepts for handling those problems.
- ◇ Section 5.2 is concerned with putting in place the performance estimation framework for large classes of first-order algorithms, objective functions, performance criteria and initial conditions. The main idea of this section is to require every element of the performance estimation problem to be

linearly Gram-representable. This section contains multiple examples of standard settings for which the methodology applies — including the settings of (sub)gradient methods (along with their projected and proximal counterparts), and conditional gradient methods.

- ◇ Finally, Section 5.3 is concerned with the application of the methodology to several concrete first-order algorithms. We provide several numerical and analytical improvements on the analysis of well-known methods, including the proximal point algorithm and the conditional gradient method.

The subsequent text is based on sections of the following preprint [THG16b].

5.1 Introduction

Consider the convex composite[1] minimization problem

$$\min_{x\in\mathbb{E}} \left\{ F(x) \equiv \sum_{k=1}^{n} F^{(k)}(x) \right\}, \tag{CM}$$

where $\mathbb{E}$ is a finite dimensional real vector space and each functional component $F^{(k)} : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$ is a convex function belonging to some class $\mathcal{F}_k(\mathbb{E})$ — e.g., smooth or non-smooth, strongly convex or not, indicator functions, etc. — for which some operations are assumed to be available in closed-form (e.g. computing a gradient, projecting on the domain, computing a proximal step, etc.).

We are interested in the composite optimization problem (CM) because it naturally allows representing and exploiting a lot of the structure in many problems, which can play a major role in our ability to efficiently solve them (see [Nes13] among others). In addition, the class of composite convex optimization problems arises very commonly in practice, as it contains for example constrained, ℓ_1 and ℓ_2-regularized convex optimization problems.

We focus on black-box oracle-based algorithms that use first-order information to approximately solve (CM), and in particular on obtaining exact and global worst-case guarantees on their performances. That is, for a given algorithm, we simultaneously seek to obtain worst-case guarantees — for example on objective function accuracy — and an instance of (CM) on which the algorithm behaves as such. In this work, we treat the case of fixed-step linear first-order methods, which includes among others fixed-step projected, proximal, conditional and inexact (sub)gradient methods.

[1] We term this objective function as *composite* because the terms may be of different natures (smooth, non-smooth, indicator, etc.); this contrasts with minimization of *finite sums* where all the terms share similar properties (see e.g., [Ber10]).

This work builds on the recent idea of performance estimation, first developed by Drori and Teboulle in [DT14] and followed-up by Kim and Fessler [KF16d] and the authors [THG16a]. The approach was initially tailored for obtaining upper bounds on the worst-case behavior of fixed-step gradient methods for unconstrained minimization of a single smooth convex objective function. Motivated by follow-up results (see among others [KF16c, KF16d]) we extend the framework of performance estimation to the composite case involving a much broader class of algorithms and function classes (see Section 1.3.1 for more details about previous works).

Our performance estimation framework relies on formulating the worst-case computation problem as a tractable semidefinite program (SDP), which can be tackled by using standard solvers [LÖ4, Mos10, Stu99]. It enjoys the following attractive features:

- ◇ any primal feasible solution to this SDP leads to a lower bound on the worst-case performance of the method under consideration, by exhibiting a particular instance of (CM),
- ◇ any dual feasible solutions to this SDP corresponds to an upper bound on the worst-case performance of the method under consideration, that can be converted into an explicit proof based on a combination of valid inequalities.

5.1.1 Performance estimation problems

In Chapter 4, we introduced a formal definition for the performance estimation problem in the case of a black-box first-order methods for unconstrained minimization of a single convex function F. We now formalize the performance estimation framework for handling multiple components in the objective function.

First, we consider black-box methods formalized using the concept of *black-box oracles*. That is, methods are only allowed to access the different components of the objective function via calling some routines returning some information about them at a given point. In particular, we focus on the standard first-order oracle for $F^{(k)}$: $\mathcal{O}_{F^{(k)}}(x) = \left\{F^{(k)}(x), \tilde{\nabla} F^{(k)}(x)\right\}$ in the sequel. The general formalism of the approach is nevertheless also valid for other standard oracles, as for examples zeroth-order or second-order ones — that is, $\mathcal{O}_{F^{(k)}}(x) = \left\{F^{(k)}(x)\right\}$ or $\mathcal{O}_{F^{(k)}}(x) = \left\{F^{(k)}(x), \nabla F^{(k)}(x), \nabla^2 F^{(k)}(x)\right\}$. However, as we will see, our ability to solve the corresponding performance estimation problems in an exact way is limited to first-order oracles at the moment.

Second, we consider a sequence of $N+1$ iterates $\{x_i\}_{0\leq i\leq N} \subset \mathbb{E}$, corresponding to a method that performs N steps from an initial iterate x_0. For each of those

iterates we consider the set of calls to the oracle for each functional component[2] $\mathcal{O}_{F^{(k)}}$: $\{\mathcal{O}_{F^{(k)}}(x_i)\}_i$. For notational convenience we denote by $K = \{1, \ldots, n\}$ the set of indices corresponding to the different components $F^{(k)}$.

Third, we consider a method $\mathcal{M}$ whose iterates can be computed by combining past and current oracle information about F. This means that after $i-1$ steps have been performed by the method, the next iterate x_i should be computable as a solution to an equation of the form:

$$\text{EQUATION}(x_0, \{\mathcal{O}_{F^{(k)}}(x_0)\}_{k\in K}, x_1, \{\mathcal{O}_{F^{(k)}}(x_1)\}_{k\in K}, \ldots, x_i, \{\mathcal{O}_{F^{(k)}}(x_i)\}_{k\in K}). \tag{5.1}$$

Note that the only unknown in this equation is x_i, and that it thus provides an implicit definition for the next step. We will see later that this assumption on $\mathcal{M}$ includes a large number of existing methods for composite optimization.

Finally, we consider a real-valued performance criterion $\mathcal{P}$, for which we assume that lower values are better. In our framework, this criterion is allowed to depend on information returned by the oracles $\mathcal{O}_{F^{(k)}}$ at all the iterates $\{x_i\}_{0\leq i\leq N}$, but also at an extra point $x_* \in \mathbb{E}$ assumed to be an optimal solution to problem (CM). The latter addition is necessary to allow criteria such the usual objective function accuracy at the last iterate $F(x_N) - F_*$ (where $F_* = F(x_*)$). We also allow the performance criterion $\mathcal{P}$ to depend on those iterates themselves, which allows for example the distance to an optimal solution $\|x_N - x_*\|^2_{\mathbb{E}}$. For notational convenience we introduce an index set for all iterates (including optimal solution) $I = \{0, 1, \ldots, N, *\}$.

The worst-case performance of method $\mathcal{M}$ on (CM) is then the optimal value of the following optimization problem, with both functions $\left\{F^{(k)}\right\}_{k\in K}$ and iterates $\{x_i\}_{i\in I}$ as variables, which we call a performance estimation problem (PEP).

$$\sup_{\left\{F^{(k)}\right\}_{k\in K}, \{x_i\}_{i\in I}} \quad \mathcal{P}(\{\mathcal{O}_{F^{(k)}}(x_i)\}_{i\in I, k\in K}, \{x_i\}_{i\in I}) \tag{PEP}$$

such that $F^{(k)} \in \mathcal{F}_k(\mathbb{E})$ for all $k \in K$,

x_0 satisfies some initialization condition,

x_{i+1} is computed by $\mathcal{M}$ according to (5.1) for all $0 \leq i \leq N-1$,

x_* is optimal for $F(x)$.

That is, a solution to (PEP) corresponds to an instance of problem (CM) on which method $\mathcal{M}$ behaves as badly as possible with respect to the performance criterion $\mathcal{P}$. The initialization condition on x_0 is required as most methods exhibit unbounded worst-case performance without it. In the sequel we will mostly restrict ourselves to the classical approach which consists in bounding

[2]That is, we chose to associate a call to each oracle to every iterate. This is mostly for notational convenience and does not induce any loss of generality. Indeed, a method can always choose not to use the information returned by one of the oracles at some iterations.

the initial distance to an optimal solution with a constant R, i.e., assume $\|x_0 - x_*\|_{\mathbb{E}} \leq R$.

Note that (PEP) is inherently an infinite-dimensional optimization problem, as functions $F^{(k)}$ appear as variables. However, a crucial observation is that, due to the black-box assumption on the objective components, this problem can be cast completely equivalently in a finite-dimensional fashion. Indeed, introducing the *outputs* of the oracle calls as variables, namely $O_i^{(k)} = \mathcal{O}_{F^{(k)}}(x_i)$ for all iterates $i \in I$ and oracles $k \in K$, we observe that steps of method $\mathcal{M}$ can be still be computed using only information contained in variables $O_i^{(k)}$, so that we can reformulate (PEP) as

$$\sup_{\left\{O_i^{(k)}\right\}_{i\in I,k\in K},\{x_i\}_{i\in I}} \mathcal{P}\left(\left\{O_i^{(k)}\right\}_{i\in I,k\in K},\{x_i\}_{i\in I}\right), \tag{PEP2}$$

such that $\exists F^{(k)} \in \mathcal{F}_k(\mathbb{E})$ satisfying $\mathcal{O}_{F^{(k)}}(x_i) = O_i^{(k)}$ for all $i \in I, k \in K$,

x_0 satisfies some initialization condition.

x_{i+1} is computed by $\mathcal{M}$ according to (5.1) for all $0 \leq i \leq N-1$,

x_* is optimal for F.

Note the central role played by the interpolation conditions, which enforce the existence of functions $F^{(k)}$ compatible with the output of the oracles. In the next subsection we describe situations for which this formulation is tractable.

5.1.2 First-order methods and convex interpolation

In the remainder of this chapter, we restrict ourselves to first-order oracles and methods. We now investigate the concept of (first-order) convex interpolability, in order to make existence constraints from (PEP2) tractable — more precise requirements are detailed in Section 5.2. From the assumptions, the existence constraint for function $F^{(k)}$

$$\exists F^{(k)} \in \mathcal{F}_k(\mathbb{E}) \text{ satisfying } \mathcal{O}_{F^{(k)}}(x_i) = O_i^{(k)} \text{ for all } i \in I,$$

found in (PEP2) may be expressed in terms of first-order information only. Considering now oracles returning first-order information only $\mathcal{O}_{F^{(k)}}(x) = \{F^{(k)}(x), \tilde{\nabla} F^{(k)}(x)\}$, we denote their output at point x_i by $\mathcal{O}_{F^{(k)}}(x_i) = O_i^{(k)} = \{f_i^{(k)}, g_i^{(k)}\}$. The above existence constraint can be rephrased into the following set of interpolation conditions

$$\exists F^{(k)} \in \mathcal{F}_k(\mathbb{E}) \text{ satisfying } F^{(k)}(x_i) = f_i^{(k)} \text{ and } \tilde{\nabla} F^{(k)}(x_i) = g_i^{(k)} \tag{INT}$$

which is exactly the concept of convex interpolation from Definition 3.1.

The notion of $\mathcal{F}$-interpolation can be considered for any class of convex functions (see Chapter 3 for more details). It allows us to formulate our performance estimation problem in its final form

$$\sup_{\left\{f_i^{(k)},g_i^{(k)}\right\}_{i\in I,k\in K},\{x_i\}_{i\in I}} \mathcal{P}\left(\left\{f_i^{(k)},g_i^{(k)}\right\}_{i\in I,k\in K},\{x_i\}_{i\in I}\right), \tag{f-PEP}$$

such that $\left\{(x_i,g_i^{(k)},f_i^{(k)})\right\}_{i\in I}$ is $\mathcal{F}_k$-interpolable for all $k\in K$,

x_0 satisfies some initialization condition.

x_{i+1} is computed by $\mathcal{M}$ according to (5.1) for all $0\leq i\leq N-1$,

x_* is optimal for F,

We conclude that identifying explicit conditions for convex interpolability by a given class of functions will be the key to eliminate the infinite-dimensional functional variables from (PEP) and transform it into a tractable estimation problem (exactly as in Chapter 4).

5.2 Performance estimation framework for first-order methods

We start this section by formulating (f-PEP) in terms of a Gram matrix. This allows obtaining a tractable convex formulation for (f-PEP) — by making appropriate assumptions on the classes of objective function components, methods, performance criteria and initialization conditions. Those assumptions are motivated by practical applications, which we also provide in the following lines. Note that the main point underlying those assumptions is to ensure that every element of the performance estimation problem can be formulated in a linear way in terms of the entries of a Gram matrix and the function values at the iterates.

5.2.1 Gram representation of iterates and objective function

Let us consider $N+1$ iterates $x_0,\ldots,x_N$ and an optimal solution x_*, and the set of corresponding oracle outputs $\{(f_i^{(k)},g_i^{(k)})\}_{i\in I,k\in K}$.

The accumulated information after those $N+1$ calls can be gathered into an $d\times(n+1)(N+2)$ matrix[3] P_N (using a slight abuse of notations) and a vector

[3]We remind the reader that $B:\mathbb{E}\to\mathbb{E}^*$ is a positive definite operator which is chosen as the identity operator in standard situations (see Section 2.1).

F_N of length $n(N+2)$:

$$P_N = [Bx_0 \ \dots \ Bx_N \mid Bx_* \mid g_0^{(1)} \ \dots \ g_0^{(n)} \mid \dots \mid g_N^{(1)} \ \dots \ g_N^{(n)} \mid g_*^{(1)} \ \dots \ g_*^{(n)}], \tag{5.2}$$

$$F_N = [\, f_0^{(1)} \ \dots \ f_0^{(n)} \mid \dots \mid f_N^{(1)} \ \dots f_N^{(n)} \mid f_*^{(1)} \ \dots f_*^{(n)}]. \tag{5.3}$$

We also denote by $B^{-1}P_N$ the matrix

$$B^{-1}P_N = [x_0 \ \dots \ x_N \mid x_* \mid B^{-1}g_0^{(1)} \ \dots \ B^{-1}g_*^{(n)}].$$

In order to formulate (PEP) in a tractable way for first-order methods, we use a Gram matrix. That is, we define the symmetric $(n+1)(N+2)\times(n+1)(N+2)$ Gram matrix $G_N \in \mathbb{S}^{(n+1)(N+2)}$, using the following construction :

$$G_N = \begin{pmatrix} \langle x_0, x_0\rangle_{\mathbb{E}} & \dots & \langle x_0, x_N\rangle_{\mathbb{E}} & \langle x_0, x_*\rangle_{\mathbb{E}} & \langle g_0^{(0)}, x_0\rangle & \dots & \langle g_*^{(n)}, x_0\rangle \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \langle x_N, x_0\rangle_{\mathbb{E}} & \dots & \langle x_N, x_N\rangle_{\mathbb{E}} & \langle x_N, x_*\rangle_{\mathbb{E}} & \langle g_0^{(0)}, x_N\rangle & \dots & \langle g_*^{(n)}, x_N\rangle \\ \langle x_*, x_0\rangle_{\mathbb{E}} & \dots & \langle x_*, x_N\rangle_{\mathbb{E}} & \langle x_*, x_*\rangle_{\mathbb{E}} & \langle g_0^{(0)}, x_*\rangle & \dots & \langle g_*^{(n)}, x_*\rangle \\ \langle g_0^{(0)}, x_0\rangle & \dots & \langle g_0^{(0)}, x_N\rangle & \langle g_0^{(0)}, x_*\rangle & \langle g_0^{(0)}, g_0^{(0)}\rangle_{\mathbb{E}^*} & \dots & \langle g_0^{(0)}, g_*^{(n)}\rangle_{\mathbb{E}^*} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \langle g_*^{(n)}, x_0\rangle & \dots & \langle g_*^{(n)}, x_N\rangle & \langle g_*^{(n)}, x_*\rangle & \langle g_*^{(n)}, g_0^{(0)}\rangle_{\mathbb{E}^*} & \dots & \langle g_*^{(n)}, g_*^{(n)}\rangle_{\mathbb{E}^*} \end{pmatrix} \succeq 0.$$

This can be written more compactly as

$$[G_N]_{ij} = \langle P_N e_i, B^{-1}P_N e_j\rangle = \langle P_N e_i, P_N e_j\rangle_{\mathbb{E}^*},$$

where $P_N e_k$ corresponds to the kth column of P_N. Also, note that the size of this matrix does not depend on the dimension d of the spaces we are working in.

Remark 5.1. Note that Gram matrix G_N is positive semidefinite for any matrix P_N (of the form (5.2)). The number of linearly independent columns of P_N is equal to the rank of G_N. Hence this rank is upper bounded by the dimension d of the ambient space of the iterates. On the other hand, it is possible to recover a matrix P_N of the form[4] (5.2) from any Gram matrix $G_N \succeq 0$ satisfying Rank $G_N \leq d$.

Our goal for the next subsections is to show that in a lot of situations, the performance estimation problem (f-PEP) can be expressed exactly as a semidefinite program in the F_N and G_N variables:

$$\sup_{G_N \succeq 0, F_N} c^\top F_N + \operatorname{Tr} C G_N \ \text{ s.t. } \ a_i + b_i^\top F_N + \operatorname{Tr} D_i G_N \leq 0 \quad \forall i \in S \tag{SDP-PEP}$$

with S some index set related to the constraints, and elements a_i, b_i, c, D_i and

[4] In the case $\mathbb{E} = \mathbb{E}^* = \mathbb{R}^d$ with the usual inner product $\langle x, y\rangle = x^\top y$ and B the identity operator, this can be done using the standard Cholesky factorization. In the general cases the exact same idea can be used, using the chosen inner product $\langle ., .\rangle_{\mathbb{E}^*}$ in the process.

C of appropriate dimensions for writing the constraints and objective function linearly in terms of the Gram matrix G_N and of the objective function values F_N.

5.2.2 Tractable formulation of the performance estimation problem

In this section, we present our main result, stating that computing the exact worst-case performance of a method on a class of functions is tractable and can, in many cases, be formulated as (SDP-PEP). We start with the concept of Gram-representability for the different ingredients of the performance estimation problem.

Definition 5.2. A class of functions is Gram-representable (resp. linearly Gram-representable) if and only if its interpolation conditions (INT) can be formulated using a finite number of convex (resp. linear) constraints involving only the matrix G_N and the corresponding function values F_N.

The functional classes of smooth strongly convex functions, smooth convex functions with bounded (sub)gradients, and strongly convex functions with bounded domain are linearly Gram-representable. In addition, the particular subclasses of support and indicator convex functions share this same advantageous property. The details and proofs of these results are postponed to Chapter 3).

Definition 5.3. A performance measure is Gram-representable (resp. linearly Gram-representable) if and only if it can be expressed as a concave (resp. linear) function involving only the matrix G_N and the corresponding function values F_N.

The class of linearly Gram-representable performance criteria contains a large variety of choices, including most standard measures we are aware of. For example, it is easy to check that standard optimality criteria in function values $F(x_N) - F(x_*)$, in residual subgradient norm $\left\|\tilde{\nabla} F(x_N)\right\|^2_{\mathbb{E}^*}$, distance to optimality $\|x_N - x_*\|^2_{\mathbb{E}}$, and distance to feasibility $\|x_N - \Pi_Q(x_N)\|^2_{\mathbb{E}}$ can be handled.

On the other hand, multiple examples of non-linear Gram-representable performance criteria can also be handled with no difficulty. This includes performance measures involving the best values among all iterates, for example $\min_{0\leq i\leq N} F(x_i) - F(x_*)$, or the best residual gradient norm among the iterates $\min_{0\leq i\leq N} \|\nabla F(x_i)\|^2_{\mathbb{E}^*}$ (see also [THG16a, Sect. 4.3] or Section 4.3.3).

Definition 5.4. An initialization condition is Gram-representable (resp. linearly Gram-representable) if and only if it can be expressed using a finite num-

ber of convex (resp. linear) constraints involving only the matrix G_N and the corresponding function values F_N.

Standard examples of valid initial conditions include the classical bounds on the initial distance to optimality $\|x_0 - x_*\|_{\mathbb{E}}^2 \leq R^2$, on the initial function value $F_0 - F_* \leq R$, and on initial gradient value $\|\nabla F(x_0)\|_{\mathbb{E}^*}^2 \leq R^2$.

Definition 5.5. A first-order method is Gram-representable (resp. linearly Gram-representable) if and only if the computation of its iterates, implicitly defined by an equation of type (5.1), can be expressed using a finite number of convex (resp. linear) constraints involving only the matrix G_N and the corresponding function values F_N.

We refer to the next section for examples of linearly Gram-representable methods.

We can now state our main results concerning Gram-representable situations. In the sequel, we recall that we use the notation $\mathcal{F}_K(\mathbb{E})$ to denote the set of functions of the form (CM) with components $F^{(k)} \in \mathcal{F}_k(\mathbb{E})$ $\forall k \in K$ — i.e., $F \in \mathcal{F}_K(\mathbb{E})$.

Proposition 5.6. Consider a class of composite objective functions $\mathcal{F}_K(\mathbb{E})$ with n components, a first-order method $\mathcal{M}$, a performance measure $\mathcal{P}$ and an initial condition $\mathcal{I}$ which are all Gram-representable.

Computing the worst-case for criterion $\mathcal{P}$ of method $\mathcal{M}$ after N iterations on objective functions in class $\mathcal{F}_K(\mathbb{E})$ with initial condition $\mathcal{I}$ can be formulated as a convex program when the dimension d of the space $\mathbb{E}$ satisfies $d \geq (n+1)(N+2)$. Otherwise, it can be formulated as a convex program plus an additional non-convex rank constraint Rank $G_N \leq d$.

If in addition $\mathcal{F}_K(\mathbb{E})$, $\mathcal{M}$, $\mathcal{P}$ and $\mathcal{I}$ are linearly Gram-representable, then the corresponding optimization problem is a SDP of the form (SDP-PEP), whose variables are $F_N \in \mathbb{R}^{n(N+2)}$ and $G_N \in \mathbb{S}^{(n+1)(N+2)}$.

Proof. The result directly follows from Remark 5.1 and from the definitions of (linear) Gram-representability for the class of functions, first-order methods, performance measures and initialization conditions: any solution to the corresponding optimization problem can be transformed into a particular instance of (CM) and vice versa. □

Remark 5.7. The optimal value of (PEP) increases with dimension d. When (PEP) with Gram-representable elements attains a finite optimal value, Proposition 5.6 implies the existence of a function with dimension at most $(n+1)(N+2)$ that achieves the worst-case value.

Remark 5.8. The assumption $d \geq (n+1)(N+2)$ is referred to as the *large-scale assumption* in the sequel. In terms of performance estimation problems, this assumption allows us to discard the non-convex rank constraint and

lead to a tractable semidefinite programming problem, which can be solved to global optimality efficiently (see e.g., [VB94]). Without that assumption, our performance estimation problem is a nonconvex rank-constrained semidefinite program, equivalent to a quadratic programming problem that is NP-hard in general (e.g., it has MAX-CUT [GW95] and other non-convex quadratic programs [PV91, Sah74] as particular cases). Approaches to handle rank constraints exist (e.g., via augmented Lagrangian techniques [BM03], via manifold optimization [JBAS10] or via Newton-like methods [OHM06]), but in general only guarantee convergence to stationary points. This is not useful in the case of (SDP-PEP), as this only provides lower bounds on the worst-case performance.

Remark 5.9. The worst-case results provided by the SDP from Proposition 5.6 provide a tight worst-case achievable for any operator B and any dual pairing $\langle .,. \rangle$.

Remark 5.10. The necessary and sufficient condition for x_* to be optimal for F is linearly Gram-representable. Indeed, it corresponds to requiring $\tilde{\nabla} F(x_*) = 0$, i.e.

$$\sum_{k\in K} \tilde{\nabla} F^{(k)}(x_*) = \sum_{k\in K} g_*^{(k)} = 0 \Leftrightarrow \left\| \sum_{k\in K} g_*^{(k)} \right\|_{\mathbb{E}^*}^2 = \langle \sum_{k\in K} g_*^{(k)}, \sum_{k\in K} g_*^{(k)} \rangle_{\mathbb{E}^*} = 0,$$

where the last condition is linear in the entries of G_N.

5.2.3 Linearly Gram-representable first-order methods

This class of first-order methods contains as particular cases what we call in the following the class of *fixed-step linear first-order methods* (FSLFOM), whose iterations are defined by a linear equation (with known constant coefficients) involving the iterates and the corresponding (sub)gradients.

Definition 5.11. A fixed-step linear first-order method (FSLFOM) is a method which computes iterate x_{i+1} as the solution of [5]

$$t_{i+1,i+1} B x_{i+1} + \sum_{k\in K} h_{i+1,i+1}^{(k)} g_{i+1}^{(k)} = \sum_{j=0}^{i} t_{i+1,j} B x_j + \sum_{j=0}^{i} \sum_{k\in K} h_{i+1,j}^{(k)} g_j^{(k)}, \tag{FSLFOM}$$

where all coefficients $h_{i+1,j}^{(k)}, t_{i+1,j} \in \mathbb{R}$ are fixed beforehand.

Note the class of FSLFOM is exactly the class of methods whose iterations

[5] Note that the iteration is written as an equality on $\mathbb{E}$, but it is possible and totally equivalent to write it on $\mathbb{E}^*$ using the operator B^{-1}, as B is invertible by assumption.

can be written in the form (using first-order optimality conditions, and the convexity of $F^{(k)}$):

$$x_{i+1} = \operatorname*{argmin}_{x\in\mathbb{E}} \left\{ \sum_{k\in K} h^{(k)}_{i+1,i+1} F^{(k)}(x) + \frac{t_{i+1,i+1}}{2}\|x\|^2_{\mathbb{E}} - \left\langle \sum_{j=1}^{i} t_{i+1,j} B x_j + \sum_{j=0}^{i}\sum_{k\in K} h^{(k)}_{i+1,j} \nabla F^{(k)}(x_j), x \right\rangle \right\};$$

which in some sense represents the most general method allowed in our framework. Those iterations can also be written by linearly combining the columns of the matrix P_N containing all the harvested first-order information about the problem:

$$0 = P_N \underline{\alpha}_k,$$

with $\underline{\alpha}_k \in \mathbb{R}^{(n+1)(N+2)}$ a vector containing appropriate coefficients. Therefore, we note that any FSLFOM is linearly Gram-representable using the following formulation:

$$0 = P_N \underline{\alpha}_k \Leftrightarrow 0 = \|P_N \underline{\alpha}_k\|^2_{\mathbb{E}^*} = \left\langle P_N \underline{\alpha}_k, B^{-1} P_N \underline{\alpha}_k \right\rangle, \tag{5.4}$$

which is clearly linear in terms of the Gram matrix G_N. Note that this can also be extended to cope with the more general class of linearly Gram-representable first-order methods[6]:

$$c^{(\mathrm{low})\top}_k F_N + b^{(\mathrm{low})}_k \leq \underline{\alpha}^\top_k G \underline{\alpha}_k \leq c^{(\mathrm{up})\top}_k F_N + b^{(\mathrm{up})}_k, \tag{5.5}$$

where $c^{(\mathrm{low})}_k, b^{(\mathrm{low})}_k$ and $c^{(\mathrm{up})}_k, b^{(\mathrm{up})}_k$ are some fixed parameters. Those can for example be used in order to require a sufficient decrease condition, or an inexact version of (FSLFOM):

$$\underline{\alpha}^\top_k G \underline{\alpha}_k \leq \varepsilon_k, \tag{Inexact FSLFOM}$$

with $\varepsilon_k \geq 0$ some accuracy parameter for the computation of (FSLFOM).

Examples of FSLFOM. Before going into the details of the performance estimation problems for our class of linear fixed-step methods and over the different classes of convex functions, let us give several examples of methods fitting into the model provided by (FSLFOM) and (Inexact FSLFOM).

- Fixed-step subgradient and gradient algorithms: fixed-step subgradient methods for minimizing a convex function F are naturally described as

[6]This formulation is just provided as an illustration to show that more general methods than (FSLFOM) can still be considered.

$x_i = x_{i-1} - \alpha_i B^{-1} g_{i-1}$ with α_i some step size, and $g_{i-1} \in \partial F(x_{i-1})$. The method is clearly in the class of FSLFOM and its linear Gram matrix representation can be obtained using formulation (5.4).

- Proximal methods and proximal gradient methods: fixed-step proximal gradient methods for minimizing $F^{(1)} + F^{(2)}$ is usually described as doing an explicit (sub)gradient step on $F^{(1)}$ followed by a minimization step on $F^{(2)}$:

$$\begin{aligned} x_i &= p_{\alpha_i F^{(2)}} \left(x_{i-1} - \alpha_i B^{-1} \tilde{\nabla} F^{(1)}(x_{i-1}) \right) \\ &= \operatorname*{argmin}_{x \in \mathbb{E}} \left\{ \alpha_i F^{(2)}(x) + \frac{1}{2} \left\| x_{i-1} - \alpha_i B^{-1} \tilde{\nabla} F^{(1)}(x_{i-1}) - x \right\|_{\mathbb{E}}^2 \right\}. \end{aligned}$$

Optimality conditions on this last term allow writing each iterations as

$$B x_i + \alpha_i \tilde{\nabla} F^{(2)}(x_i) = B x_{i-1} - \alpha_i \tilde{\nabla} F^{(1)}(x_{i-1}),$$

with some $\tilde{\nabla} F^{(2)}(x_i) \in \partial F^{(2)}(x_i)$. This method is clearly a FSLFOM and therefore fits in the framework. Also, note that projected gradient methods are obtained using the same technique, but on the particular class of convex indicator functions, whereas proximal point algorithms correspond to the case where $F^{(1)} = 0$.

- Conditional gradient methods do also fit into the model provided by Equation (FSLFOM). Indeed, the iterations take the following form:

$$\begin{aligned} y_i &= \operatorname*{argmin}_{z \in \mathbb{E}} \left\{ \left\langle z - z_i, \tilde{\nabla} F^{(1)}(z_i) \right\rangle + F^{(2)}(z) \right\}, \\ z_{i+1} &= (1 - \lambda_i) z_i + \lambda_i y_i, \end{aligned}$$

with $\lambda_i \in [0, 1]$ chosen beforehand. Now, by imposing y_i using first-order necessary and sufficient optimality conditions on the intermediate optimization problem, we obtain

$$\tilde{\nabla} F^{(1)}(z_i) = -\tilde{\nabla} F^{(2)}(y_i).$$

Note that for conditional gradient-type methods, $F^{(2)}$ is usually chosen as the indicator function of some closed convex set Q. This algorithm can also clearly be written as a FSLFOM by artificially denoting for $i = 0, 1, \ldots$ the iterates $x_{2i} = z_i$ and $x_{2i+1} = y_i$.

- Inexact (sub)gradient methods for a convex function $F^{(1)}$, with $x_{i+1} = x_i - \alpha_i B^{-1}(\tilde{\nabla} F^{(1)}(x_i) + \varepsilon_i)$ and $\|\varepsilon_i\|_{\mathbb{E}^*} \leq \epsilon_i$ for some $\epsilon_i \geq 0$ the tolerance on the (sub)gradient computation. This can be written in the inexact FSLFOM format:

$$\left\| \alpha_i \left(x_{i+1} - x_i \right) + \tilde{\nabla} F^{(1)}(x_i) \right\|_{\mathbb{E}^*}^2 \leq \epsilon_i^2.$$

Also, note that other noise models, as for example the one proposed by d'Aspremont [d'A08] can also easily be used in the framework. On the other hand, the inexact (δ, L)-oracles developed by Devolder et al. [DGN14] do not seem to easily fit into the approach[7].

Note that a broad class of methods can be modelled using those operations, just by requiring the functions on which it is applied to belong to certain classes. As an example, alternate projection-type algorithms are special cases of proximal methods, applied on the class of convex indicator functions. Therefore, they can be represented in the FSLFOM format.

5.2.4 Simplified performance estimation problems

Note that for standard algorithms such as the previous examples of FSLFOM, the SDP resulting from Proposition 5.6 can typically be further simplified, leading to a reduction in its size.

Corollary 5.12. Consider a class of functions $\mathcal{F}_K(\mathbb{E})$, a performance measure $\mathcal{P}$ and an initialization condition $\mathcal{I}$ which are linearly Gram-representable, and a FSLFOM $\mathcal{M}$ whose iterations are all linearly independent[8].

In addition, assume there are p points $(g_i^{(k)}, f_i^{(k)})$ such that neither $g_i^{(k)}$ nor $f_i^{(k)}$ are used in the performance measure $\mathcal{P}$, the initial condition $\mathcal{I}$ and the method $\mathcal{M}$. Then, the performance estimation problem can be written as a convex SDP using variables $F_N \in \mathbb{R}^{n(N+2)-p}$ and $G_N \in \mathbb{S}^{(n+1)(N+2)-N-p-1}$, with the possible additional rank constraint rank $G_N \leq d$.

To see why this corollary holds, note that the variables in the simplified SDP correspond to the function values and the Gram matrix from which the p unnecessary points were removed, and from which N other variables were substituted using the N iteration constraints (FSLFOM)[9].

Under the assumptions of Corollary 5.12, the large-scale assumption becomes $d \geq (n+1)(N+2) - N - p - 1$. In the cases where only the output from a single oracle is used at each iteration, we have that $p = (n-1)(N+1)$, which leads to $d \geq N + n + 2$.

[7] This is due to the fact no necessary and sufficient interpolation conditions for this noise model were found — that is, standard conditions are only necessary to guarantee interpolability. Using necessary conditions that are not sufficient still allows obtaining upper bounds on the worst-case behavior, but those may not be tight.

[8] That is, the vectors $\underline{\alpha}_k$ used to characterize the iterations are linearly independent — this is very reasonable, as every method using new information at each iteration satisfies this. Remark that it does not imply that the points x_i themselves are linearly independent.

[9] The additional -1 term appearing in the dimension of the Gram matrix comes from the fact that one of the $g_*^{(k)}$ may also be discarded, by substituting it using the optimality condition of x_*.

Furthermore, for standard performance measures (e.g. $F_N - F_*$, $\|x_N - x_*\|_{\mathbb{E}}^2$, $\left\|\tilde{\nabla} F(x_N)\right\|_{\mathbb{E}^*}^2$), one arbitrary point x_i may be fixed to 0 because solutions to the SDP are be invariant with respect to translations. This would result in the large-scale assumption $d \geq N + n + 1$. For $n = 1$, we recover the standard $d \geq N + 2$ appearing in the case of a single component in the objective function (see Theorem 4.2 and Corollary 4.5).

The original SDP from Proposition 5.6 may be challenging to solve in practice, because of its potentially large size on the one hand, and because its feasible region may lack an interior on the other hand. We observe that the simplified performance estimation problem described above typically improves the situation for both issues, reducing the size of the problem and solving in a lot of cases the issue of a lack of interior points. Finally, note that as in the case of unconstrained optimization, structural properties of (SDP-PEP) could potentially be used for further simplifying the SDP (see Remark 4.6).

5.3 Algorithm analysis

In this section, we analytically and numerically study different algorithms for solving variants of (CM), and compare our results with standard guarantees from the literature[10]. This section is organized as follows:

- We begin with an analytical study of a proximal point algorithm. For this algorithm, we provide simple analytical and tight convergence results twice better than the standard theoretical guarantees. We also illustrate how to incorporate a simple noise model into the performance estimation framework for this basic method.
- Secondly, we numerically study several simple variants of projected subgradient methods; this illustrates the applicability of (PEP) for very simple and widely studied methods.
- Third, we use performance estimation to compare several standard variants of fast proximal gradient methods. An tentative extension to the optimized gradient method (OGM) proposed by Kim and Fessler [KF16d] is proposed using the ideas developed for fast proximal gradient methods.
- Finally, we conclude by illustrating the results of the approach for a conditional gradient and on two alternate projections schemes. Those choices illustrates the applicability of the approach for studying a large variety of methods and performance measures.

[10]Note that most of the literature results are presented for B being the identity operator (and hence $\mathbb{E} = \mathbb{E}^*$). We will nevertheless compare our slightly more general results with the standard bounds from the literature (thus even when they are officially valid only for B being the identity) — we recall that our results are valid for general self-adjoint positive definite linear operator $B : \mathbb{E} \to \mathbb{E}^*$ (see Remark 5.9).

5.3.1 A proximal point algorithm

Consider a simple model with only one convex (possibly non-smooth) term in the objective function,

$$\min_{x\in\mathbb{E}} F(x),$$

with $F \in \mathcal{F}_{0,\infty}(\mathbb{E})$. In this first example, we assume that the proximal operation is easy to compute for F:

$$x_{k+1} = p_{\alpha_{k+1}F}(x_k) = \operatorname*{argmin}_{x\in\mathbb{E}} \left\{ \alpha_{k+1}F(x) + \frac{1}{2}\|x_k - x\|_{\mathbb{E}}^2 \right\}.$$

That is, the iterations can be written in the form of an implicit method $x_{k+1} = x_k - \alpha_{k+1}B^{-1}g_{k+1}$, for some $g_{k+1} \in \partial F(x_{k+1})$. For recent overviews and motivations concerning proximal algorithms, we refer the reader to the work of Combettes and Pesquet[11] [CP11], and to the review works of Bertsekas [Ber10] and Parikh and Boyd [PB13]. For historical point of view on those methods, we refer to the pioneer works of Moreau [Mor65], Rockafellar [Roc76] and the analysis of Guler [Gül91].

Proximal Point Algorithm (PPA)

Input: $F \in \mathcal{F}_{0,\infty}(\mathbb{E})$, $x_0 \in \mathbb{E}$. Parameters: $\{\alpha_k\}_k$ with $\alpha_k \geq 0$.

For $k = 1 : N$

$$x_k = p_{\alpha_k F}(x_{k-1})$$

Convergence of PPA in function and gradient values

The standard convergence result for the proximal point algorithm is provided by Guler in [Gül91, Theorem 2.1] :

$$F(x_N) - F_* \leq \frac{R^2}{2\sum_{k=1}^N \alpha_k},$$

for any initial condition x_0 satisfying $\|x_0 - x_*\|_{\mathbb{E}} \leq R$. We improve this bound by a factor 2 using the PEP approach.

Theorem 5.13. Let $\{\alpha_k\}_k$ be a sequence of positive step sizes and x_0 some initial iterate satisfying $\|x_0 - x_*\|_{\mathbb{E}} \leq R$ for some optimal point x_*. Any sequence $\{x_k\}_k$ generated by the proximal point algorithm with step sizes $\{\alpha_k\}_k$

[11] This work features among others a large list of known proximal operators.

on a function $F \in \mathcal{F}_{0,\infty}(\mathbb{E})$ satisfies

$$F(x_N) - F_* \leq \frac{R^2}{4\sum_{k=1}^N \alpha_k}.$$

In addition, this bound is tight, and it is attained on the l_1-shaped one-dimensional function ($\dim \mathbb{E} = \dim \mathbb{E}^* = 1$) $F(x) = \frac{\sqrt{B}R|x|}{2\sum_{k=1}^N \alpha_k} = \frac{R\|x\|_{\mathbb{E}}}{2\sum_{k=1}^N \alpha_k}$ with $Bx_0^2 = R^2$.

Proof. The proof relies on finding a primal and dual form of (f-PEP) for the proximal point algorithm (details can be found in Appendix 5.A). A dual solution allows us to obtain the upper bound part. □

Considering another convergence measure, the exact same idea allows us to obtain strong numerical evidence for the following conjecture.

Conjecture 5.14. Let $\{\alpha_k\}_k$ be a sequence of positive step sizes and x_0 some initial iterate satisfying $\|x_0 - x_*\|_{\mathbb{E}} \leq R$ for some optimal point x_*. For any sequence $\{x_k\}_k$ generated by the proximal point algorithm with step sizes $\{\alpha_k\}_k$ on a function $F \in \mathcal{F}_{0,\infty}(\mathbb{E})$, there exists a subgradient $g_N \in \partial F(x_N)$ such that

$$\|g_N\|_{\mathbb{E}^*} \leq \frac{R}{\sum_{k=1}^N \alpha_k}.$$

In particular, the choice $g_N = \frac{Bx_{N-1} - Bx_N}{\alpha_N}$ is a subgradient satisfying the inequality.

Observe that this bound cannot be improved, as it is attained on the (one-dimensional) l_1-shaped function $F(x) = \frac{\sqrt{B}R|x|}{\sum_{k=1}^N \alpha_k}$ with $Bx_0^2 = R^2$. The particular choice of subgradient suggested in the theorem corresponds to the subgradient appearing in the proximal operation when written as an implicit subgradient step.

This sort of convergence results in terms of residual (sub)gradient norm is particularly interesting when considering dual methods. In that case, the dual residual gradient norm corresponds to the primal distance to feasibility (see e.g., [DGN12]).

PPA with a basic uncertainty model

Assume that one can approximately compute the proximal operation $x_k = x_{k-1} - \alpha_k B^{-1}(g_k - g_k^{(\varepsilon)})$ with the guarantee that

$$\left\| \frac{x_k - x_{k-1}}{\alpha_k} + B^{-1} g_k \right\|_{\mathbb{E}} = \left\| g_k^{(\varepsilon)} \right\|_{\mathbb{E}^*} \leq \epsilon, \tag{5.6}$$

with $\epsilon \geq 0$ a given precision for iteration k and $\alpha_k \geq 0$ the step size at iteration k. Even though this model is very basic, it can for example be used when it is possible to approximate $g_{k+1} \approx g_k$ with a controlled error $\|g_k - g_{k+1}\|_{\mathbb{E}^*} \leq \epsilon$, in which case the explicit algorithm $x_k = x_{k-1} - \alpha_k B^{-1} g_{k-1}$ satisfies the assumptions. More practically, it can also be used when the proximal operation can only approximately be solved — that is, the proximal operator can only approximately be evaluated, with the guarantee that:

$$\exists \tilde{\nabla} F(x_k) \in \partial F(x_k) \text{ s.t. } \left\| \frac{x_k - x_{k-1}}{\alpha_k} + B^{-1} \tilde{\nabla} F(x_k) \right\|_{\mathbb{E}} \leq \epsilon.$$

The following theorem illustrates the type of results that can be obtained using the PEP approach for this uncertain PPA.

Theorem 5.15. Let $\{\alpha_k\}_k$ be a sequence of positive step sizes, x_0 some initial iterate satisfying $\|x_0 - x_*\|_{\mathbb{E}} \leq R$ for some optimal point x_* and $\epsilon > 0$ be a positive constant. There exists a function $F \in \mathcal{F}_{0,\infty}(\mathbb{E})$ such that the sequence $\{x_i\}_i$ generated by the proximal point algorithm with step sizes $\{\alpha_k\}_k$ under the noise model (5.6) satisfies

$$F(x_N) - F_* \geq R\epsilon,$$

for any x_0 satisfying $\|x_0 - x_*\|_{\mathbb{E}} \leq R$ for some optimal point x_*.

In addition, the case $\sum_{k=1}^{N} \alpha_k = \frac{R}{\epsilon}$ leads to $F(x_N) - F_* \leq R\epsilon$.

Proof. The proof is presented in Appendix 5.B, and also relies on finding a lower bound and a matching dual feasible point. □

Note the choice of step sizes $\{\alpha_k\}_k$ satisfying $\sum_{k=1}^{N} \alpha_k = \frac{R}{\epsilon}$ is optimal for that setting, as its worst-case guarantee match the lower bound

$$F(x_N) - F_* \geq R\epsilon,$$

that is valid for any choice of step sizes.

We leave further investigations in this direction as future work.

5.3.2 Projected subgradient methods

In this section, we consider the constrained non-smooth convex optimization problem

$$\min_{x \in Q} F(x), \tag{5.7}$$

with $F \in \mathcal{C}_{M,\infty}(\mathbb{E})$ and $Q \subseteq \mathbb{E}$ a closed convex set. For solving this kind of problems, we assume here on the one hand that for any $x \in Q$, one can easily compute at least one element $g \in \partial F(x)$, and on the other hand that one can easily compute projections onto the set Q.

In this setting, one can use the projected subgradient method in order to obtain approximations to x_*. That is, one can iterate $x_{k+1} = \Pi_Q(x_k - \alpha_{k+1} B^{-1} g_k)$, with $\Pi_Q(.)$ the projection operator onto Q, with $g_k \in \partial f(x_k)$ and with $\alpha_k \geq 0$ some step size parameters.

Projected Subgradient Method (PSM)

Input: $F \in \mathcal{C}_{M,\infty}(\mathbb{E})$, $x_0 \in Q \subseteq \mathbb{E}$.

For $k = 1 : N$

$$x_k = \Pi_Q \left(x_{k-1} - \alpha_i B^{-1} g_{k-1} \right)$$

In that setting, it is known that for every step size policy $\{\alpha_i\}_i$, there exists a function such that both (e.g., [DT16, Theorem A.1][12]):

$$\min_{0 \leq i \leq N} F(x_i) - F_* \geq \frac{MR}{\sqrt{N+1}}, \quad F\left(\frac{1}{N+1} \sum_{i=0}^{N} x_i \right) - F_* \geq \frac{MR}{\sqrt{N+1}}, \tag{5.8}$$

(we refer to the first criterion as the performance of the best iterate, and to the second criterion as the performance of the averaged iterate). In fact, the simple constant step size policy $\alpha_k = \frac{R}{M\sqrt{N+1}}$ allows obtaining the corresponding tight worst-case guarantees (see e.g., [Bub15, Nes04]):

$$\min_{0 \leq i \leq N} F(x_i) - F_* \leq \frac{MR}{\sqrt{N+1}}, \quad F\left(\frac{1}{N+1} \sum_{i=0}^{N} x_i \right) - F_* \leq \frac{MR}{\sqrt{N+1}}. \tag{5.9}$$

However, this constant step size policy is very impractical, as its use require the knowledge of the number of iterations in advance. Therefore, we use the performance estimation framework to refine the guarantees that can be obtained with the more practical step size policies $\alpha_k = \frac{R}{M} \frac{1}{\sqrt{k+1}}$ and $\alpha_k = \frac{R}{M} \frac{1}{k+1}$.

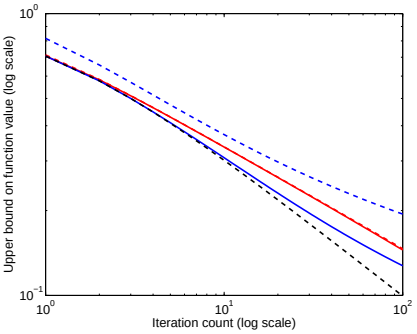
[12] Note that [DT16, Theorem A.1] does not directly treat those cases, but the function it uses in the proof remains valid for them.

As a reference, we use the following standard guarantee (e.g., [BXM03, Nes04]):

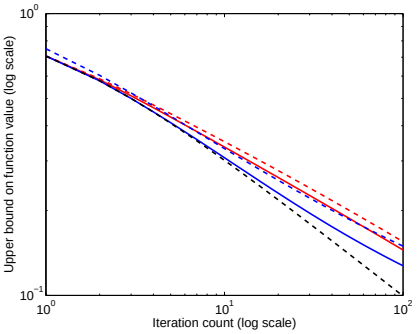
$$\min_{0\leq i\leq N} F(x_i) - F_* \leq \frac{R^2 + M^2 \sum_{k=0}^{N} \alpha_k^2}{2 \sum_{k=0}^{N} \alpha_k}. \tag{5.10}$$

First, we compare the performances of the different step size policies for the best iterates[13] on Figure 5.1(a). Those results indicates that the step sizes $\alpha_k = \frac{R}{M}\frac{1}{k+1}$ should a priori be preferred over $\alpha_k = \frac{R}{M}\frac{1}{\sqrt{k+1}}$ for low number of iterations, which is exactly opposite to what is advised by the standard guarantee (5.10).

Second, we compare the worst-case performance of the best iterate with the worst-case performance of the averaged iterate[14] on Figure 5.1(b). Interestingly we observe that better performances should be expected from using the best iterate, and not the averaged one — we are note aware of a generic guarantee like (5.10) for the averaged iterate.



(a) Comparison between the worst-case guarantees of PEP with the theoretical bound (5.10) for the best iterate. Step sizes $\alpha_k = \frac{R}{M}\frac{1}{k+1}$, theoretical guarantee (5.10) (dashed, blue), PEP guarantee (plain, blue); step sizes $\alpha_k = \frac{R}{M}\frac{1}{\sqrt{k+1}}$, theoretical guarantee (5.10) (dashed, red), PEP guarantee (plain, red) and lower bound (5.8) (dashed, black).



(b) Comparison between the worst-case performances of the last iterate and of the averaged iterate (using performance estimation). Step sizes $\alpha_k = \frac{R}{M}\frac{1}{k+1}$, averaged iterate (dashed, blue), best iterate (plain, blue); step sizes $\alpha_k = \frac{R}{M}\frac{1}{\sqrt{k+1}}$, averaged iterate (dashed, red), best iterate (plain, red) and lower bound (5.8) (dashed, black).

Figure 5.1: Projected subgradient method: comparison between the results of PEP with the theoretical bound (5.10) for the best iterate, and comparison between convergence in the best iterate and convergence in the averaged iterate.

[13] That is, the worst-case performance for the criterion $\min_{0\leq i\leq N} F(x_i) - F_*$.

[14] That is, the worst-case performance for the criterion $F\left(\frac{1}{N+1}\sum_{i=0}^{N} x_i\right) - F_*$.

5.3.3 Fast proximal gradient algorithms

In this section, we consider the two-terms composite objective function

$$\min_{x\in\mathbb{E}} \left\{ F(x) \equiv F^{(1)}(x) + F^{(2)}(x) \right\}, \tag{5.11}$$

with $F^{(1)} \in \mathcal{F}_{0,L}(\mathbb{E})$ (smooth convex function) and $F \in \mathcal{F}_{0,\infty}(\mathbb{E})$ (non-smooth convex function). We assume that gradients are easy to compute for $F^{(1)}$, and that the proximal operation is easy to compute for $F^{(2)}$:

$$p_{\alpha F^{(2)}}(x) = \operatorname*{argmin}_{y\in\mathbb{E}} \left\{ \alpha F(y) + \frac{1}{2}\|x-y\|_{\mathbb{E}}^2 \right\}.$$

In order to approximatively solve (5.11), it is common to use different variants of fast proximal gradient methods (FPGM). We numerically investigate the worst-case guarantees of two variants using different step sizes policies, and propose new variants with slightly better worst-case behaviors. Also, we illustrate differences in the worst-case performances obtained in the cases where $F^{(2)} = 0$ (unconstrained smooth convex minimization), $F^{(2)} \in \mathcal{I}_{\infty}(\mathbb{E})$ (constrained smooth convex minimization) or $F^{(2)} \in \mathcal{F}_{0,\infty}(\mathbb{E})$ (regularized smooth convex minimization).

In the following, we call FPGM1 the standard fast proximal gradient method (FISTA [BT09b]), FPGM2 a variant with slightly better guarantees, and POGM a proximal version of the optimized gradient method [KF16d]. FPGM2 and POGM illustrate how performance estimation problems can be used in the development of new optimization algorithms ; their study in this chapter remains however entirely numerical.

Standard Fast Proximal Gradient Methods (FPMG1)

The first variants of accelerated proximal methods we are considering use a standard proximal step after an explicit gradient step for generating the so-called *primary sequence* $\{y_k\}_k$.

Fast Proximal Gradient Method (FPGM1)

Input: $F^{(1)} \in \mathcal{F}_{0,L}(\mathbb{E})$, $F^{(2)} \in \mathcal{F}_{0,\infty}(\mathbb{E})$ $x_0 \in \mathbb{E}$, $y_0 = x_0$.

For $k = 1 : N$

$$y_k = p_{F^{(2)}/L}\left(x_{k-1} - \frac{1}{L} B^{-1} \nabla F^{(1)}(x_{k-1}) \right)$$
$$x_k = y_k + \alpha_k (y_k - y_{k-1})$$

In this algorithm, we refer to α_k as *inertial parameters*. We use two standard variants: $\alpha_k^{(a)} = \frac{k-1}{k+2}$ — among others proposed in [SBC14, Tse08] — and

$\alpha_k^{(b)} = \frac{\theta_{k-1}-1}{\theta_k}$ (with $\theta_k = \frac{1+\sqrt{4\theta_{k-1}^2+1}}{2}$ and $\theta_0 = 1$) — see [BT09b, Nes83, Tse08]. For both variants, the standard convergence result is (see e.g., [BT09b, SBC14]):

$$F(y_N) - F_* \leq \frac{2LR^2}{(N+1)^2}, \tag{5.12}$$

for any initial iterate x_0 such that $\|x_0 - x_*\|_{\mathbb{E}}^2 \leq R^2$. We numerically compare those two variants of FPGM1 using (f-PEP) on Figure 5.3.3. After 100 iterations, both inertial parameter policies perform about the same way (parameters $\alpha_k^{(b)}$ performs only about 2% better than $\alpha_k^{(a)}$ in terms of worst-case performances). We also observe that the behavior of both variants of FPGM1 is well captured by the standard guarantee (5.12).

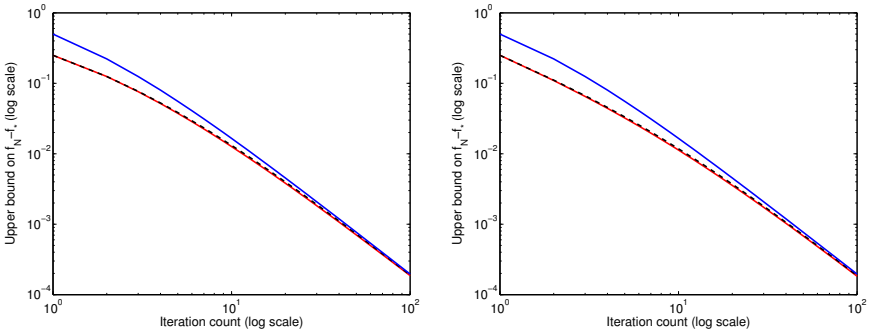


Figure 5.2: Comparison of the worst-case convergence speed of the different variants of FPGM1 (left) and FPGM2 (right) for $N \in \{1, \ldots, 100\}$, $L = 1$ and $R = 1$. The curves respectively corresponds to the different inertial coefficient, namely $\alpha_k^{(a)}$ (dashed, black) and $\alpha_k^{(b)}$ (red), and the standard guarantee (5.12) (blue).

New Fast Proximal Gradient Methods (FPGM2)

Secondary sequences $\{x_k\}$ are usually converging slightly faster than primary sequences $\{y_k\}$ in the unconstrained case ($F^{(2)} = 0$), as observed in [KF16d, THG16a]. However, some issues may arise with the secondary sequences of FPGM1 when applied to constrained or proximal problems: iterates may in some cases become infeasible, or the objective may become unbounded (see Table 5.1 below). We therefore propose new variants of fast proximal gradient methods that do not suffer from theses drawback, called FPGM2 (with two different step size policies). Part of the underlying motivation behind FPGM2 is also the ability to generalize it later to the optimized gradient method.

Remark 5.16. The design of FPGM2 is based on two ideas: on the one hand, it should be equivalent to the standard fast gradient method in the case of smooth unconstrained convex minimization, and on the other hand, it should not move after two consecutive iterates have reached the same optimal point for (5.11) (i.e., $x_{k-1} = x_k = x_*$ implies $x_{k+1} = x_*$).

Fast Proximal Gradient Method 2 (FPGM2)

Input: $F^{(1)} \in \mathcal{F}_{0,L}(\mathbb{E})$, $F^{(2)} \in \mathcal{F}_{0,\infty}(\mathbb{E})$ $x_0 \in \mathbb{E}$, $z_0 = y_0 = x_0$.

For $k = 1 : N$

$$y_k = x_{k-1} - \frac{1}{L} B^{-1} \nabla F^{(1)}(x_{k-1})$$
$$z_k = y_k + \alpha_k (y_k - y_{k-1}) + \frac{\alpha_k}{L\gamma_{k-1}} (z_{k-1} - x_{k-1})$$
$$x_k = p_{\gamma_k F^{(2)}}(z_k)$$

In this algorithm, we use the coefficients $\gamma_k = \frac{\alpha_k + 1}{L}$. Note that we introduced two intermediate sequences: on the one hand sequence $\{\gamma_k\}_k$, corresponding to the step sizes to be taken by the proximal steps, and on the other hand sequence $\{z_k\}_k$, which allows keeping track of the subgradient used in the proximal steps (note that $\frac{1}{\gamma_k}(z_k - x_k)$ corresponds to the subgradient used in the proximal step from z_k to x_k). Even if FPGM2 may look more intricate than the classical FPGM1, it is in fact simpler, as it involves only one sequence on which both implicit (proximal) and explicit (gradient) steps are being taken. Indeed, explicit steps are taken using gradient values of $F^{(1)}$ at x_k, and subgradients used in the proximal steps are subgradients of $F^{(2)}$ also at x_k. This can be seen by rewriting the iterations of FPGM2 using the secondary sequence $\{x_k\}_k$ only

$$\begin{aligned} x_{k+1} =& x_k + \alpha_{k+1}(x_k - x_{k-1}) \\ &+ \frac{\alpha_{k+1}}{L} B^{-1} \nabla F^{(1)}(x_{k-1}) - \frac{1}{L} B^{-1} \nabla F^{(1)}(x_k) - \frac{\alpha_{k+1}}{L} B^{-1} \nabla F^{(1)}(x_k) \\ &+ \frac{\alpha_{k+1}}{L} B^{-1} \tilde{\nabla} F^{(2)}(x_k) - \frac{1}{L} B^{-1} \tilde{\nabla} F^{(2)}(x_{k+1}) - \frac{\alpha_{k+1}}{L} B^{-1} \tilde{\nabla} F^{(2)}(x_{k+1}), \end{aligned}$$

with $\tilde{\nabla} F^{(2)}(x_k)$ the subgradient of $F^{(2)}$ used in the proximal operation generating x_k.

Comparing the different variants of FPGM2 on Figure 5.3.3 leads to the same conclusion as for FPGM1: inertial parameters $\alpha^{(b)}$ perform slightly better than $\alpha^{(a)}$.

In Table 5.1, we report the different worst-case performances guarantees obtained numerically for FPGM1 (for both sequences) and FPGM2 (for the better secondary sequence only). We consider three situations: $F^{(2)} = 0$ (unconstrained smooth convex minimization), $F^{(2)} \in \mathcal{I}_\infty(\mathbb{E})$ (constrained smooth

convex minimization with projected methods) and $F^{(2)} \in \mathcal{F}_{0,\infty}(\mathbb{E})$ (regularized smooth convex minimization with proximal methods).

Type	$F(y_N) - F_*$ (FPGM1)	$F(x_N) - F_*$ (FPGM1)	$F(x_N) - F_*$ (FPGM2)
Unconstrained ($F^{(2)} = 0$)	$\frac{LR^2}{2}\frac{4}{N^2+5N+6}$	$\frac{LR^2}{2}\frac{4}{N^2+7N+4}$	$\frac{LR^2}{2}\frac{4}{N^2+7N+4}$
Constrained ($F^{(2)} \in \mathcal{I}_\infty$)	$\frac{LR^2}{2}\frac{4}{N^2+5N+2}$	Infeasible	$\frac{LR^2}{2}\frac{4}{N^2+7N}$
Non-smooth ($F^{(2)} \in \mathcal{F}_{0,\infty}$)	$\frac{LR^2}{2}\frac{4}{N^2+5N+2}$	Unbounded	$\frac{LR^2}{2}\frac{4}{N^2+7N}$

Table 5.1: Worst-case obtained for FPGM1 and FPGM2 with inertial coefficient $\alpha_k = \frac{k-1}{k+2}$ and $N \geq 1$.

Convergence results reported in the table correspond to properly identified functions (i.e. they are rigorous lower bounds). After solving the corresponding PEPs numerically (for $L = R = 1$ and $1 \leq N \leq 100$), we conjecture them to be equal to the exact worst-case guarantees.

We observe that the worst-case guarantees for FPGM2 are slightly better than for FPGM1.Guarantees for the unconstrained case are slightly better than those for the constrained and proximal cases, which are equal. Note that the secondary sequence of FPGM1 is not guaranteed to be feasible in the constrained case, and that the corresponding objective value may be unbounded in the proximal case (for any $N \geq 1$).

The worst-case functions identified numerically for the unconstrained case are Huber-shaped functions [THG16a]. In the constrained case, we identified one-dimensional linear optimization problems of the form $\min_{x \geq 0} cx$ as worst-cases, where c is a constant defined by

$$c = \frac{\sqrt{B}R}{2\sum_{i=0}^{N-1} [\underline{\alpha}_N]_i} .$$

Finally, for the proximal case, our worst-case has function $F^{(1)}(x) = cx$ with the same c as above, and function $F^{(2)}(x)$ may be chosen equal to zero for $x \geq 0$ and to sx for $x < 0$, for any negative value of the slope $s < 0$.

5.3.4 A proximal optimized gradient method

In this section, we consider again the regularized smooth convex minimization problem (5.11). In particular, we are concerned with the possibility of obtaining

optimized methods for handling this sort of problems (i.e., methods whose worst-case performances are minimized).

The idea is to extend the optimized gradient method (OGM) developed by Kim and Fessler in [KF16d], which is originally tailored for smooth unconstrained minimization ($F^{(2)} = 0$). In the unconstrained smooth minimization setting, this first-order method was recently shown in [Dro16] to have the best achievable worst-case guarantee for the criterion $F_N - F_*$.

The method we propose has been obtained by combining the ideas obtained from the original OGM [KF16d] and the non-standard placement of the proximal operator used for speeding up the convergence of fast proximal gradient methods (FPGM2). It was designed using the same two principles as FPGM2 (see Remark 5.16): on the one hand, it is equivalent to OGM when applied to smooth unconstrained convex minimization problems, and on the other hand, it remains at an optimal point when it reaches one.

Proximal Optimized Gradient Method (POGM)

Input: $F^{(1)} \in \mathcal{F}_{0,L}(\mathbb{E})$, $F^{(2)} \in \mathcal{F}_{0,\infty}(\mathbb{E})$, $x_0 \in \mathbb{E}$, $y_0 = x_0$, $\theta_0 = 1$.

For $k = 1 : N$

$$y_k = x_{k-1} - \frac{1}{L} B^{-1} \nabla F^{(1)}(x_{k-1})$$

$$z_k = y_k + \frac{\theta_{k-1} - 1}{\theta_k}(y_k - y_{k-1}) + \frac{\theta_{k-1}}{\theta_k}(y_k - x_{k-1}) + \frac{\theta_{k-1} - 1}{L\gamma_{k-1}\theta_k}(z_{k-1} - x_{k-1})$$

$$x_k = p_{\gamma_k F^{(2)}}(z_k)$$

In this algorithm, we use the sequence $\gamma_k = \frac{1}{L}\frac{2\theta_{k-1}+\theta_k-1}{\theta_k}$ and the inertial coefficients proposed in [KF16d]:

$$\theta_k = \begin{cases} \frac{1+\sqrt{4\theta_{k-1}^2+1}}{2}, & i \leq N-1 \\ \frac{1+\sqrt{8\theta_{k-1}^2+1}}{2}, & i = N \end{cases}$$

Simply trying to generalize OGM using the standard proximal step on the primary sequence $\{y_i\}$ (as for FPGM1) does not lead to a converging algorithm. We have numerical evidence, i.e. worst-case functions showing that this candidate algorithm does not see its worst-case improving after each iteration: in other words its worst-case rate is not converging to zero). Therefore we have to introduce the same idea used in FPGM2 concerning the place of the proximal operator.

We compare POGM to FPGM1 and FPGM2 with inertia $\alpha_k^{(b)}$ on Figure 5.3. We obtain worst-case performances about twice better for POGM when compared to both FPGM1 and FPGM2. Of course, POGM suffers from the drawback of requiring the knowledge of the number of iterations in advance (this is because

the rule to compute the last coefficient θ_N differs from the rule to compute all the previous ones). This practical disadvantage is not easily solved: if the last θ_N is updated with the same rule as all the previous θ_k, performance is degraded by a non-negligible factor, rendering it even slower than FPGM (note that this is already the case for smooth unconstrained minimization [KF16c]).

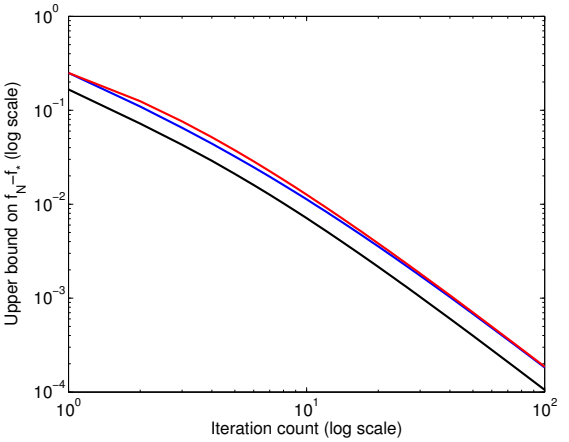


Figure 5.3: Comparison between the worst-case performances of FPGM1 (with inertia $\alpha_k^{(b)}$) (red), FPGM2 (with inertia $\alpha_k^{(b)}$) (blue), POGM (black) and OGM (dashed, black) for $N \in \{1, \dots, 100\}$, $L = 1$ and $R = 1$. The worst-case performances of POGM are about twice better than the worst-case performance of FPGM between 1 and 100 iterations. Also, we observe that OGM [KF16d] (equivalent to POGM with $F^{(2)} = 0$) behaves approximately 12% better than POGM in the worst-case.

5.3.5 A conditional gradient method

Consider the constrained smooth convex optimization problem

$$\min_{x \in Q} F(x),$$

with $F \in \mathcal{F}_{0,L}(\mathbb{E})$ and $Q \subset \mathbb{E}$ a bounded and closed convex set. In that setting, there exists different ways for treating the constraint set Q. In the previous section, we proposed to use fast gradient methods, which require the ability of projecting onto the closed convex set Q. In this section, we rather consider the standard conditional gradient method (also sometimes referred to as the Frank-Wolfe method), which originates from [FW56]. This algorithm has the advantage of not requiring to perform projections onto Q, but rather to perform

linear optimization on this set (which is typically easier when Q is a polyhedral set).

Conditional Gradient Method (CGM)

Input: $F \in \mathcal{F}_{0,L}(\mathbb{E})$, closed convex $Q \subset \mathbb{E}$ with $\|x-y\|_{\mathbb{E}} \leq D \ \forall x, y \in Q$, $x_0 \in Q$.

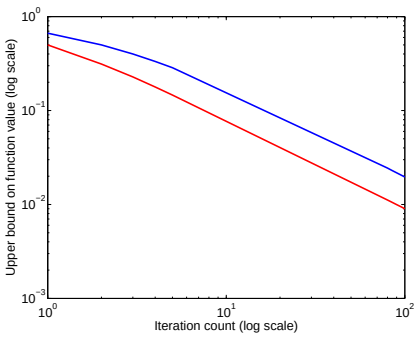
For $k = 1 : N$

$$y_k = \operatorname*{argmin}_{y \in Q} \{\langle \nabla F(x_{k-1}), y - x_{k-1} \rangle\}$$
$$\lambda_k = \frac{2}{1+k}$$
$$x_k = (1-\lambda_k) x_{k-1} + \lambda_k y_k$$

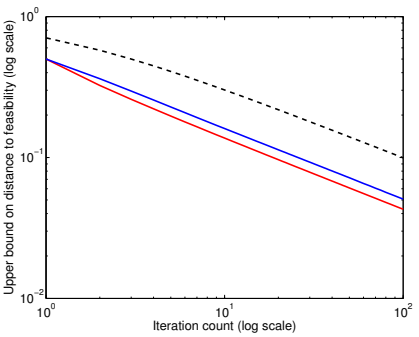
The standard global convergence guarantee for this method (see e.g., [Jag13, Theorem 1]) is

$$F(x_N) - F_* \leq \frac{2LD^2}{N+2}, \tag{5.13}$$

which we compare with the exact bound provided by PEP on Figure 5.4(a). As illustrated in Section 5.2.3, this algorithm fits into the (FSLFOM) format. The numerical guarantees we obtained by solving the performance estimation problem are between two and three times better than the standard guarantee, depending on the number of iterations.



(a) Worst-case performance of CGM (red) and its theoretical guarante (5.13) (blue) for $N \in \{1, \ldots, 100\}$, $L = 1$ and $D = 1$.



(b) Worst-case performance of APM (red), DAPM (blue) and lower bound $\frac{MR}{\sqrt{N+1}}$ for subgradient methods (dashed, black) for $N \in \{1, \ldots, 100\}$ and $R = 1$ ($M = 1$ by definition of the objective function (5.14)).

Figure 5.4: Numerical analysis of a conditional gradient method (left) and of two variants of alternate projections algorithms (right).

5.3.6 Alternate projection and Dykstra methods

In this section, we numerically investigate the difference between the worst-case behaviors of the standard alternate projection method (APM) for finding a point in the intersection of two convex sets, and the Dykstra [BD86] method (DAPM) for finding the closest point in the intersection of two convex sets. APM is known to converge sublinearly in general, as it is a particular instance of subgradient-type descent[15] applied to the problem

$$\min_{x\in\mathbb{E}}\{f(x)=\max_i \|x-\Pi_{Q_i}(x)\|_{\mathbb{E}}\}, \tag{5.14}$$

whose objective function is convex and non-smooth (with Lipschitz constant $M=1$). Therefore, its expected global convergence rate is $\mathcal{O}(\frac{1}{\sqrt{N}})$ (see [DT16, Theorem A.1]). We compare below the convergence of both APM and DAPM with the standard lower bound for subgradient schemes $\frac{MR}{\sqrt{N+1}}$ as a reference.

Alternate Projection Method (APM)

Input: $x_0 \in \mathbb{E}$, convex sets $Q_1, Q_2 \subseteq \mathbb{E}$, $\|x_0 - x_*\|_{\mathbb{E}} \leq R$, for some $x_* \in Q_1 \cap Q_2$.

For $k = 1 : N$

$$x_k = \Pi_{Q_2}(\Pi_{Q_1}(x_{k-1}))$$

Dykstra Alternate Projection Method (DAPM)

Input: $x_0 \in \mathbb{E}$, convex sets $Q_1, Q_2 \subseteq \mathbb{E}$, $\|x_0 - x_*\|_{\mathbb{E}} \leq R$, for some $x_* \in Q_1 \cap Q_2$. Initialize $p_0 = q_0 = 0$.

For $k = 0 : N-1$

$$\begin{aligned} y_k &= \Pi_{Q_1}(x_k + p_k)\\ p_{k+1} &= x_k + p_k - y_k\\ x_{i+1} &= \Pi_{Q_2}(y_k + q_k)\\ q_{k+1} &= y_k + q_k - x_{k+1} \end{aligned}$$

The optimality measure used is $\min_{x\in Q_1}\|x - x_N\|_{\mathbb{E}} = \|\Pi_{Q_1}(x_N) - x_N\|_{\mathbb{E}}$ (note that $x_N \in Q_2$). We do not give further details the corresponding performance estimation problem here, as it is very similar to the previous sections. The

[15] It can be shown that $\frac{x-\Pi_{Q_k}(x)}{||x-\Pi_{Q_k}(x)||}$ is a subgradient of the function $f(x)$ (at x such that $f(x) = ||x - \Pi_{Q_k}(x)||$). Therefore, in the case of two sets Q_1, Q_2, and assuming that x is feasible for one of the two sets (say, Q_1), a projection on the other one corresponds to a subgradient step on f with step size $||x-\Pi_{Q_2}(x)||$. Hence, APM is an instance of subgradient method for $k > 1$ (when x_k is feasible for one of the two sets).

results for APM and DAPM are shown on Figure 5.4(b), where the (expected) convergence in $\mathcal{O}(\frac{1}{\sqrt{N}})$ is clearly obtained. Interestingly, DAPM converges slightly slower than APM (more precisely, DAPM has a worst-case about 18% higher than APM), which is therefore more advisable in terms of worst-case performances for finding a point in the intersection of two convex sets when no additional structure is assumed. In addition, note that both APM and DAPM have a worst-case which is about twice better than the standard lower bound for explicit non-smooth schemes.

5.4 Conclusion

In this chapter, we presented a performance estimation approach for analyzing first-order algorithm for composite optimization problems. The results of Chapter 4 (or [THG16a]) were largely extended to handle both larger classes of objective functions (components) and larger classes of first-order algorithms to possibly be analyzed, all that in a more general setting for handling pairs of conjugate norms. The contribution is essentially twofold: first, we exploit the structures of interpolation conditions from Chapter 3 to formulate the exact worst-case problem for fixed-step linear first-order methods with appropriate convergence measure and initial conditions. Secondly, we apply the methodology to provide tight analyses for different first-order methods.

Further extensions to the performance estimation framework are proposed in Chapter 8, including for coping with randomness and monotone operators.

Software. MATLAB implementations of the performance estimation approach for different variants of gradient methods are available online. It can be downloaded from `http://perso.uclouvain.be/adrien.taylor`.

Appendix

5.A Proof of Theorem 5.13

We start by proving the lower bound, and then we prove the matching upper bound.

Lower bound. Let us show that applying PPA to the one-dimensional function $F(x) = \frac{\sqrt{B}R|x|}{2\sum_{k=1}^{N}\alpha_k}$ with $x_0 = -\frac{R}{\sqrt{B}}$ allows us to achieve:

$$F(x_N) - F(x_*) = \frac{R^2}{4\sum_{k=1}^{N}\alpha_k},$$

which shows that the bound from Theorem 5.13 is tight.

First, note that for $x \neq 0$, we have $\nabla F(x) = \operatorname{sign}(x)\frac{\sqrt{B}R}{2\sum_{k=1}^{N}\alpha_k}$. Hence,

$$x_N = x_0 + B^{-1}\sum_{k=1}^{N}\alpha_k\frac{\sqrt{B}R}{2\sum_{k=1}^{N}\alpha_k} = -\frac{R}{2\sqrt{B}}.$$

Therefore, by noting that $x_* = 0$ and $F(x_*) = 0$, we have the desired result.

Upper bound. In order to express the corresponding PEP in the simplest form, we heavily rely on some straightforward simplifications of (SDP-PEP) (see Corollary 5.12 and Remark 5.2.4). Let us denote by P_N the matrix containing the information harvested after N iterations (we use the notation g_i for subgradients $g_i \in \partial F(x_i)$): $P_N = [g_1\ g_2\ \ldots\ g_N\ Bx_0]$, and by G_N its corresponding Gram matrix (see Section 5.2.1). Also, we introduce the step size matrices $\underline{\alpha}_k$ for expressing all intermediate iterates x_i's in terms of x_0 and the subgradient g_i's, that is: $x_k = P_N\underline{\alpha}_k \quad (k = 0, \ldots, N)$.

This results in the following explicit expressions for $\underline{\alpha}_k$: $\underline{\alpha}_k = e_{N+1} - \sum_{i=1}^{k}\alpha_i e_i$, along with $\underline{\alpha}_0 = e_{N+1}$ and $\underline{\alpha}_* = 0$ (i.e., we assume without loss of generality

$x_* = 0$), where we use the standard notation e_i for the unit vector having a single 1 as its i^{th} component — we also denote $e_* = 0$. In order to perform the wort-case analysis for PPA, we now formulate the performance estimation problem (f-PEP) as the following SDP (simplified version of (SDP-PEP) where the x_k's ($k = 1, \dots, N$) have been substituted using the form of the algorithm $x_k = x_{k-1} - \alpha_k B^{-1} g_k$):

$$\begin{aligned} \max_{G_N \in \mathbb{S}^{N+1}, f_1, \dots, f_N, f_* \in \mathbb{R}^N} \; & f_N - f_*, \qquad \text{(PPA-PEP)} \\ \text{s.t. } & f_j - f_i + \operatorname{Tr}(A_{ij} G_N) \leq 0, \quad i, j \in \{1, \dots, N, *\} \\ & \|x_0 - x_*\|_{\mathbb{E}}^2 \leq R^2, \\ & G_N \succeq 0, \end{aligned}$$

with $2A_{ij} = e_j(\underline{\alpha}_i - \underline{\alpha}_j)^\top + (\underline{\alpha}_i - \underline{\alpha}_j)e_j^\top$, the matrices coming from the non-smooth convex interpolation inequalities (see Condition (3.1)).

In order to obtain an analytical upper bound for PPA, we consider the Lagrangian dual to (PPA-PEP), which is given by the following.

$$\begin{aligned} \min_{\lambda_{ij} \geq 0, \tau \geq 0} \; & \tau R^2 \qquad \text{(PPA-dPEP)} \\ \text{s.t. } & e_N - \sum_i \sum_{j \neq i} (\lambda_{ij} - \lambda_{ji}) e_j = 0, \\ & \sum_i \sum_{j \neq i} \lambda_{ij} A_{ij} + \tau \underline{\alpha}_0 \underline{\alpha}_0^\top \succeq 0, \end{aligned}$$

(where the constraint corresponding to f_* can be discarded since it is clear that letting $f_* = 0$ does not change the optimal solution of (PPA-PEP)). Note that the set of equality constraints can be assimilated to a set of *flow* constraints on a complete directed graph. That is, considering a graph where the optimum and each iterate correspond to nodes, each $0 \leq \lambda_{ij} \leq 1$ corresponds to the flow on the edge going from node j to node i (we choose this direction by convention). This flow constraint imposes that the outgoing flow equals the ingoing flow for every node, except at iterate N where the outgoing flow should be equal to 1 and at the optimum, where the incoming flow should be equal to 1. We show that the following choice is a feasible point of the dual (PPA-dPEP).

$$\lambda_{i,i+1} = \frac{\sum_{k=1}^{i} \alpha_k}{2\sum_{k=1}^{N} \alpha_k - \sum_{k=1}^{i} \alpha_k}, \qquad i \in \{1, \dots, N-1\}$$

$$\lambda_{*,i} = \frac{2\alpha_i \sum_{k=1}^{N} \alpha_k}{\left(2\sum_{k=1}^{N} \alpha_k - \sum_{k=1}^{i} \alpha_k\right)\left(2\sum_{k=1}^{N} \alpha_k - \sum_{k=1}^{i-1} \alpha_k\right)}, \qquad i \in \{1, \dots, N\}$$

$$\tau = \frac{1}{4\sum_{k=1}^{N} \alpha_k},$$

and $\lambda_{ij} = 0$ otherwise. First, we clearly have $\lambda_{ij} \geq 0$ and some basic computations allow to verify that the equality constraints from (PPA-dPEP) are satisfied:

$$\begin{aligned}
\lambda_{*,1} - \lambda_{1,2} &= 0,\\
&\vdots\\
\lambda_{*,i} + \lambda_{i-1,i} - \lambda_{i,i+1} &= 0,\\
&\vdots\\
\lambda_{*,N-1} + \lambda_{N-2,N-1} - \lambda_{N-1,N} &= 0,\\
\lambda_{*,N} + \lambda_{N-1,N} &= 1.
\end{aligned}$$

It remains to show that the corresponding dual matrix S is positive semidefinite.

$$\begin{aligned}
2S = &\sum_{i=1}^{N-1} 2\alpha_{i+1}\lambda_{i,i+1} e_{i+1}e_{i+1}^\top + 2\tau e_{N+1}e_{N+1}^\top\\
&+ \sum_{i=1}^{N} \lambda_{*,i}\left[e_i\left(-e_{N+1} + \sum_{k=1}^{i}\alpha_k e_k\right)^\top + \left(-e_{N+1} + \sum_{k=1}^{i}\alpha_k e_k\right)e_i^\top\right].
\end{aligned}$$

In order to reduce the number of indices to be used, we note shortly $\lambda_i = \lambda_{i,i+1}$ and $\mu_i = \lambda_{*,i}$. Then, using the equality constraints, we arrive at the following dual matrix:

$$2S = \begin{pmatrix}
2\alpha_1\lambda_1 & \alpha_1\mu_2 & \alpha_1\mu_3 & \dots & \alpha_1\mu_{N-1} & \alpha_1\mu_N & -\mu_1\\
\alpha_1\mu_2 & 2\alpha_2\lambda_2 & \alpha_2\mu_3 & \dots & \alpha_2\mu_{N-1} & \alpha_2\mu_N & -\mu_2\\
\alpha_1\mu_3 & \alpha_2\mu_3 & 2\alpha_3\lambda_3 & \dots & \alpha_3\mu_{N-1} & \alpha_3\mu_N & -\mu_3\\
\vdots & & \ddots & \ddots & & \vdots & \vdots\\
\alpha_1\mu_{N-1} & \alpha_2\mu_{N-1} & \alpha_3\mu_{N-1} & \dots & 2\alpha_{N-1}\lambda_{N-1} & \alpha_{N-1}\mu_N & -\mu_{N-1}\\
\alpha_1\mu_N & \alpha_2\mu_N & \alpha_3\mu_N & \dots & \alpha_{N-1}\mu_N & 2\alpha_N & -\mu_N\\
-\mu_1 & -\mu_2 & -\mu_3 & \dots & -\mu_{N-1} & -\mu_N & 2\tau
\end{pmatrix}.$$

In order to prove $S \succeq 0$, we first use a Schur complement and then show that the resulting matrix is diagonally dominant with positive diagonal elements.

After the Schur complement, we obtain the matrix $\tilde{S}$:

$$\tilde{S} = \begin{pmatrix} 2\alpha_1\lambda_1 & \alpha_1\mu_2 & \alpha_1\mu_3 & \dots & \alpha_1\mu_{N-1} & \alpha_1\mu_N \\ \alpha_1\mu_2 & 2\alpha_2\lambda_2 & \alpha_2\mu_3 & \dots & \alpha_2\mu_{N-1} & \alpha_2\mu_N \\ \alpha_1\mu_3 & \alpha_2\mu_3 & 2\alpha_3\lambda_3 & \dots & \alpha_3\mu_{N-1} & \alpha_3\mu_N \\ \vdots & & \ddots & \ddots & & \vdots \\ \alpha_1\mu_{N-1} & \alpha_2\mu_{N-1} & \alpha_3\mu_{N-1} & \dots & 2\alpha_{N-1}\lambda_{N-1} & \alpha_{N-1}\mu_N \\ \alpha_1\mu_N & \alpha_2\mu_N & \alpha_3\mu_N & \dots & \alpha_{N-1}\mu_N & 2\alpha_N \end{pmatrix} - \frac{1}{2\tau} \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_N \end{pmatrix} \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_N \end{pmatrix}^\top.$$

The first step to show the diagonally dominant character of $\tilde{S}$ is to note that every non-diagonal element of $\tilde{S}$ is non-positive: $\alpha_j\mu_i - \frac{\mu_i\mu_j}{2\tau} \leq 0, \quad \forall i \neq j$. Indeed, this is equivalent to write this in the following form ($\mu_i > 0$):

$$\begin{aligned} &\alpha_j - \frac{\mu_j}{2\tau} \\ &= \alpha_j \left(\frac{\left(2\sum_{k=1}^N \alpha_k - \sum_{k=1}^i \alpha_k\right)\left(2\sum_{k=1}^N \alpha_k - \sum_{k=1}^{i-1} \alpha_k\right) - \left(2\sum_{k=1}^N \alpha_k\right)^2}{\left(2\sum_{k=1}^N \alpha_k - \sum_{k=1}^i \alpha_k\right)\left(2\sum_{k=1}^N \alpha_k - \sum_{k=1}^{i-1} \alpha_k\right)} \right) \leq 0, \end{aligned}$$

since $\alpha_k \geq 0$ by assumption. This allows us to discard the absolute values in the diagonally dominance criteria. Then, using the equality constraints, we obtain an expression for the sum of all non-diagonal elements of line i of $\tilde{S}$:

$$\begin{aligned} \mu_i \sum_{j=1}^{i-1} \alpha_j + \alpha_i \sum_{j=i+1}^{N} \mu_j - \frac{\mu_i}{2\tau} \sum_{j \neq i} \mu_j \\ = \begin{cases} \mu_i \sum_{j=1}^{i-1} \alpha_j + \alpha_i(1-\lambda_i) - \frac{1}{2\tau}\mu_i(1-\mu_i), & \text{if } i < N \\ \mu_N \sum_{j=1}^{N-1} \alpha_j - \frac{1}{2\tau}\mu_N(1-\mu_N) & \text{if } i = N \end{cases} \end{aligned}$$

Using the values of μ_i, λ_i and τ along with elementary computations allow to verify $\forall i \in \{1, \dots, N\}$:

$$\begin{cases} -(\mu_i \sum_{j=1}^{i-1} \alpha_j + \alpha_i(1-\lambda_i) - \frac{1}{2\tau}\mu_i(1-\mu_i)) & = 2\alpha_i\lambda_i - \frac{\mu_i^2}{2\tau} \quad \text{if } i = 1, \dots, N-1, \\ -(\mu_i \sum_{j=1}^{i-1} \alpha_j - \frac{1}{2\tau}\mu_i(1-\mu_i)) & = 2\alpha_i - \frac{\mu_i^2}{2\tau} \quad \text{if } i = N, \end{cases}$$

which implies diagonal dominance of $\tilde{S}$ (even more: the sum of the elements of each line equals 0). □

5.B Proof of Theorem 5.15

Again, we denote by P_N the matrix containing the information required to model the algorithm in the (SDP-PEP) format:

$$P_N = [g_1 \ g_2 \ \dots \ g_N \ g_1^{(\epsilon)} \ g_2^{(\epsilon)} \ \dots \ g_N^{(\epsilon)} \ Bx_0],$$

and by G_N its corresponding Gram matrix (see Section 5.2.1). In that case, the step size vectors $\underline{\alpha}_k$ are defined as

$$x_k = P_N \underline{\alpha}_k \quad (k = 0, \dots, N),$$

and can be written in the following form:

$$\underline{\alpha}_k = e_{2N+1} - \sum_{i=1}^{k} \alpha_i (e_i + e_{N+i})$$

along with $\underline{\alpha}_* = 0$ (again, we use the standard notation e_i for the unit vector having a single 1 as its i^{th} component and we denote $e_* = 0$). We can now state the performance estimation problem of the inexact PPA in the (SDP-PEP) format:

$$\begin{aligned} \max_{G_N \in \mathbb{S}^{N+1}, F_N \in \mathbb{R}^N} \ & f_N - f_* && \text{(uncPPA-PEP)} \\ \text{s.t. } & f_j - f_i + \operatorname{Tr}(A_{ij} G_N) \leq 0, && i,j \in \{1, \dots, N, *\} \\ & \left\| g_i^{(\epsilon)} \right\|_{\mathbb{E}^*}^2 \leq \epsilon^2, && i \in \{1, \dots, N\} \\ & \|x_0 - x_*\|_{\mathbb{E}}^2 \leq R^2, \\ & G_N \succeq 0, \end{aligned}$$

with $2A_{ij} = e_j(\underline{\alpha}_i - \underline{\alpha}_j)^\top + (\underline{\alpha}_i - \underline{\alpha}_j)e_j^\top$, coming again from the non-smooth interpolation constraints (see Theorem 3.4).

In order to obtain an analytical upper bounds for PPA, we consider the Lagrangian dual to (uncPPA-PEP), which is given by the following.

$$\begin{aligned} \min_{\lambda_{ij} \geq 0, \gamma_{ij} \geq 0, \tau \geq 0} \ & \tau R^2 + \epsilon^2 \sum_{i=1}^{N} \gamma_i && \text{(uncPPA-dPEP)} \\ & \text{s.t. } e_N - \sum_i \sum_{j \neq i} (\lambda_{ij} - \lambda_{ji}) e_j = 0, \\ & \sum_i \sum_{j \neq i} \lambda_{ij} A_{ij} + \sum_i \gamma_i e_{N+i} e_{N+i}^\top + \tau \underline{\alpha}_0 \underline{\alpha}_0^\top \succeq 0. \end{aligned}$$

Lower bound. We start by considering the following one-dimensional lower bound: $F(x) = \epsilon\sqrt{B}|x|$, and $g^{(\epsilon)} = -\epsilon\sqrt{B}$. Clearly, whatever the starting point $x_0 > 0$, the algorithm does not allow it to move (similar conclusion for $g^{(\epsilon)} = \epsilon\sqrt{B}$ and $x_0 < 0$). By choosing $x_0 = R/\sqrt{B}$, we have the desired lower bound.

Upper bound. Then, we show that the bound is actually tight with $\sum_{k=1}^{N} \alpha_k = \frac{R}{\epsilon}$ by showing that the following choice is dual feasible:

$$\tau = \frac{1}{2\sum_{k=1}^{N} \alpha_k}, \quad \lambda_{i,i+1} = \frac{\sum_{k=1}^{i} \alpha_k}{\sum_{k=1}^{N} \alpha_k}, \quad \lambda_{*,i} = \frac{\alpha_i}{\sum_{k=1}^{N} \alpha_k}, \quad \gamma_i = \frac{\alpha_i}{2}.$$

Those conditions satisfy the flow constraints, as very few computations allow to verify:

$$\begin{aligned}
\lambda_{1,2} - \lambda_{*,1} &= 0,\\
\lambda_{2,3} - \lambda_{1,2} - \lambda_{*,2} &= 0,\\
&\vdots\\
\lambda_{N-1,N} - \lambda_{N-2,N-1} - \lambda_{*,N-1} &= 0,\\
\lambda_{N-1,N} + \lambda_{*,N} &= 1.
\end{aligned}$$

Let us show that the corresponding dual matrix S is positive semidefinite.

$$\begin{aligned}
2S = &\sum_{i=1}^{N-1} \alpha_{i+1}\lambda_{i,i+1}(2e_{i+1}e_{i+1}^\top + e_{i+1}e_{N+i+1}^\top + e_{N+i+1}e_{i+1}^\top)\\
&+ \sum_{k=1}^{N} 2\gamma_i e_{N+k}e_{N+k}^\top + 2\tau e_{2N+1}e_{2N+1}^\top\\
&+ \sum_{i=1}^{N} \lambda_{*,i}\left[e_i\left(-e_{2N+1} + \sum_{k=1}^{i}\alpha_k(e_k + e_{N+k})\right)^\top\right.\\
&\qquad \left. + \left(-e_{2N+1} + \sum_{k=1}^{i}\alpha_k(e_k + e_{N+k})\right)e_i^\top\right].
\end{aligned}$$

As in the proof of Theorem 5.13, we use a Schur complement followed by a weak diagonal dominance to prove positive semidefinitess of S. The Schur

complement allows us to write:

$$\tilde{S} = \begin{pmatrix} 2\alpha_1\lambda_1 & \alpha_1\mu_2 & \alpha_1\mu_3 & \dots & \alpha_1\mu_{N-1} & \alpha_1\mu_N \\ \alpha_1\mu_2 & 2\alpha_2\lambda_2 & \alpha_2\mu_3 & \dots & \alpha_2\mu_{N-1} & \alpha_2\mu_N \\ \alpha_1\mu_3 & \alpha_2\mu_3 & 2\alpha_3\lambda_3 & \dots & \alpha_3\mu_{N-1} & \alpha_3\mu_N \\ \vdots & & \ddots & \ddots & & \vdots \\ \alpha_1\mu_{N-1} & \alpha_2\mu_{N-1} & \alpha_3\mu_{N-1} & \dots & 2\alpha_{N-1}\lambda_{N-1} & \alpha_{N-1}\mu_N \\ \alpha_1\mu_N & \alpha_2\mu_N & \alpha_3\mu_N & \dots & \alpha_{N-1}\mu_N & 2\alpha_N \end{pmatrix} - LD^{-1}L^\top,$$

with the lower triangular matrix L and the diagonal matrix D respectively defined as

$$L = \begin{pmatrix} \alpha_1\lambda_1 & 0 & 0 & \dots & 0 & -\mu_1 \\ \alpha_1\mu_2 & \alpha_2\lambda_2 & 0 & \dots & 0 & -\mu_2 \\ \vdots & & & \ddots & & \vdots \\ \alpha_1\mu_N & \alpha_2\mu_N & \dots & & \alpha_{N-1}\lambda_{N-1} & -\mu_N \end{pmatrix},$$

$$D = 2\begin{pmatrix} \gamma_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & \gamma_2 & 0 & \dots & 0 & 0 \\ \vdots & & & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & & \gamma_N & 0 \\ 0 & 0 & \dots & & 0 & \tau \end{pmatrix}.$$

Before going into the details, let us denote the matrix $\tilde{S} = A - B$ with A and B the matrices arising previously from the Schur complement argument on $2S$. The elements of A are

$$a_{ij} = \begin{cases} \frac{\alpha_i\alpha_j}{\sum_{k=1}^N \alpha_k} & \text{if } i \neq j \\ \frac{2\alpha_i \sum_{k=1}^i \alpha_k}{\sum_{k=1}^N \alpha_k} & \text{otherwise.} \end{cases}$$

The elements of B are

$$b_{ij} = \begin{cases} \frac{2\alpha_i\alpha_j \sum_{k=1}^{j-1} \alpha_k + \alpha_i\alpha_j^2}{\left(\sum_{k=1}^N \alpha_k\right)^2} + \frac{\alpha_i\alpha_j}{\sum_{k=1}^N \alpha_k} & \text{if } i \neq j, i > j, \\ \frac{\alpha_i^2\left(\sum_{k=1}^N \alpha_k + \sum_{k=1}^{i-1} \alpha_k\right) + \alpha_i\left(\sum_{k=1}^i \alpha_k\right)^2}{\left(\sum_{k=1}^N \alpha_k\right)^2} & \text{if } i = j, \\ b_{ji} & \text{otherwise.} \end{cases}$$

The following computations allow to guarantee that the diagonal terms of $\tilde{S}$ are indeed non-negative:

$$\begin{aligned} a_{ii} &= \frac{2\alpha_i \left(\sum_{k=1}^{i} \alpha_k\right) \left(\sum_{k=1}^{N} \alpha_k\right)}{\left(\sum_{k=1}^{N} \alpha_k\right)^2}, \\ &= \frac{\alpha_i^2 \left(\sum_{k=1}^{N} \alpha_k\right) + \alpha_i \left(\sum_{k=1}^{i-1} \alpha_k\right) \left(\sum_{k=1}^{N} \alpha_k\right)}{\left(\sum_{k=1}^{N} \alpha_k\right)^2}, \\ &\quad + \frac{\alpha_i \left(\sum_{k=1}^{i} \alpha_k\right)^2 + \alpha_i \left(\sum_{k=1}^{i} \alpha_k\right) \left(\sum_{k=i+1}^{N} \alpha_k\right)}{\left(\sum_{k=1}^{N} \alpha_k\right)^2} \\ &\geq b_{ii}, \end{aligned}$$

using the non-negativity of h_k. Some computations In order to discard the absolute values in the diagonal dominance criterion, remark that all off-diagonal terms of $\tilde{S}$ are non-positive, as for $i \neq j$, we have

$$\tilde{S}_{ij} = a_{ij} - b_{ij} = -\frac{2\alpha_i \alpha_j \sum_{k=1}^{j-1} \alpha_k + \alpha_i \alpha_j^2}{\left(\sum_{k=1}^{N} \alpha_k\right)^2}.$$

In order to conclude the proof, we show that $\tilde{S}$ is weakly diagonally dominant, as we have

$$\tilde{S}_{ii} = \sum_{j=1}^{i-1} (-\tilde{S}_{ij}) + \sum_{j=i+1}^{N} (-\tilde{S}_{ji}).$$

This can be written in terms of the elements of $\tilde{S}$, and by using the values of the multipliers as

$$\begin{aligned} &\frac{\alpha_i}{\left(\sum_{k=1}^{N} \alpha_k\right)^2} \left[\left(\sum_{k=1}^{i} \alpha_k \right) \left(\sum_{k=1}^{N} \alpha_k + \sum_{k=i+1}^{N} \alpha_k \right) - \alpha_i \sum_{k=1}^{N} \alpha_k - \alpha_i \sum_{k=1}^{i-1} \alpha_k \right] \\ &= \frac{\alpha_i}{\left(\sum_{k=1}^{N} \alpha_k\right)^2} \left[\sum_{j=1}^{i-1} \left(\alpha_j \sum_{k=1}^{j} \alpha_k \right) + \sum_{j=1}^{i-1} \sum_{k=1}^{j-1} \alpha_j \alpha_k + \sum_{j=i+1}^{N} \alpha_j \left(\sum_{k=1}^{i} \alpha_k + \sum_{k=1}^{i-1} \alpha_k \right) \right]. \end{aligned}$$

Basic simplifications allow to show that proving the last equality is equivalent to prove the following:

$$\sum_{k=1}^{i-1} \alpha_k \left(\sum_{j=1}^{N} \alpha_j - \alpha_i \right) = \sum_{j=1}^{i-1} \alpha_j \left(\sum_{k=1}^{j} \alpha_k + \sum_{k=1}^{j-1} \alpha_k + \sum_{k=i+1}^{N} \alpha_k \right).$$

To prove this equality, we focus on the right-hand term:

$$\begin{aligned}
&\sum_{j=1}^{i-1} \alpha_j \left(\sum_{k=1}^{j} \alpha_k + \sum_{k=1}^{j-1} \alpha_k + \sum_{k=i+1}^{N} \alpha_k \right) \\
&\qquad = \sum_{k=1}^{i-1} \sum_{j=1}^{k} \alpha_j \alpha_k + \sum_{k=1}^{i-2} \sum_{j=k+1}^{i-1} \alpha_j \alpha_k + \sum_{k=1}^{i-1} \sum_{j=i+1}^{N} \alpha_j \alpha_k, \\
&\qquad = \sum_{k=1}^{i-2} \sum_{j=1}^{i-1} \alpha_j \alpha_k + \sum_{j=1}^{i-1} \alpha_j \alpha_{i-1} + \sum_{k=1}^{i-1} \sum_{j=i+1}^{N} \alpha_j \alpha_k, \\
&\qquad = \sum_{k=1}^{i-1} \sum_{j=1}^{i-1} \alpha_j \alpha_k + \sum_{k=1}^{i-1} \sum_{k=i+1}^{N} \alpha_j \alpha_k, \\
&\qquad = \sum_{k=1}^{i-1} \alpha_k \left(\sum_{j=1}^{N} \alpha_j - \alpha_i \right).
\end{aligned}$$

In order to conclude the proof, observe that the objective value of the dual is

$$\tau R^2 + \epsilon^2 \sum_{i=1}^{N} \gamma_i = R\epsilon,$$

using the multiplier values along with $\sum_{k=1}^{N} \alpha_k = \frac{R}{\epsilon}$. □

Chapter 6

Steepest Descent with Exact Line Search

In this chapter, we consider the gradient (or steepest) descent method with exact line search applied to a strongly convex function with Lipschitz continuous gradient. We establish the exact worst-case rate of convergence of this scheme, and show that this worst-case behavior is exhibited by a certain convex quadratic function. We also give the tight worst-case complexity bound for a noisy variant of gradient descent method, where exact line search is performed in a search direction that differs from negative gradient by at most a prescribed relative tolerance.

The main contributions of the chapter are the following:

- ◇ we provide a tight analysis of the steepest descent algorithm with exact line search for when the function to be minimized is smooth and strongly convex.
- ◇ We provide a tight analysis of an inexact version of steepest descent with exact line search on the same class of functions.

This chapter is divided into five sections:

- ◇ in Section 6.1, we introduce the steepest descent algorithm with exact line search, claim the main results and compare them with the literature.
- ◇ Section 6.2 summarizes properties of steepest descent with exact line search used prove our main results. Also, the corresponding performance estimation problem is formulated.
- ◇ In Section 6.3, we prove our main result using the performance estimation approach.

- ◇ In Section 6.4, we perform a similar analysis for an inexact version. The methodology used in the analysis relies on performance estimations problems in the same way as in the exact case.
- ◇ Finally, Section 6.5 presents our concluding remarks on the algorithms and on the methodology.

The subsequent text is based on the paper [dKGT16].

Note that for the sake of simplicity, we work in the case $\mathbb{E} = \mathbb{E}^* = \mathbb{R}^d$ in the following pages, but all the results also hold for different primal and dual Euclidean spaces (see Section 2.1 and Chapter 5).

6.1 Introduction

The gradient (or steepest) descent method for unconstrained method was devised by Augustin-Louis Cauchy (1789–1857) in the 19th century, and remains one of the most iconic algorithms for unconstrained optimization. Indeed, it is usually the first algorithm that is taught during introductory courses on nonlinear optimization. It is therefore somewhat surprising that the worst-case convergence rate of the method is not yet precisely understood for smooth strongly convex functions.

In this chapter we settle the worst-case convergence rate question of the gradient descent method with exact line search for strongly convex, continuously differentiable functions f with Lipschitz continuous gradient.

The gradient method with exact line search may be described as follows.

Gradient descent method with exact line search

Input: $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, $x_0 \in \mathbb{R}^d$.

for $i = 0, 1, \ldots$

$$\gamma = \operatorname*{argmin}_{\gamma \in \mathbb{R}} f\left(x_i - \gamma \nabla f(x_i)\right)$$
$$x_{i+1} = x_i - \gamma \nabla f(x_i)$$

Our main result may now be stated concisely.

Theorem 6.1. Let $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, x_* a global minimizer of f on $\mathbb{R}^d$, and $f_* = f(x_*)$. Each iteration of the gradient method with exact line search satisfies

$$f(x_{i+1}) - f_* \leq \left(\frac{L-\mu}{L+\mu}\right)^2 (f(x_i) - f_*) \quad i = 0, 1, \ldots \tag{6.1}$$

Note that the result in Theorem 6.1, which establishes a global linear convergence rate on objective function accuracy, is known for the case of quadratic

functions in $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$, that is for functions of the form

$$f(x) = \frac{1}{2}x^\top Qx + c^\top x$$

where $c \in \mathbb{R}^d$, and the eigenvalues of the $n \times n$ symmetric positive definite matrix Q lie in the interval $[\mu, L]$; see e.g. [Ber99, §1.3], [Pol87, pp. 60–62], or [LY08, pp. 235–238]. Moreover, the bound (6.1) is known to be tight for the following example.

Example 6.2. Consider the following quadratic function from [Ber99, Example on p.69]:

$$f(x) = \frac{1}{2}x^\top \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_d)\, x$$

where

$$0 < \mu = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_d = L,$$

and the starting point

$$x_0 = (\frac{1}{\mu}, 0, \ldots, 0, \frac{1}{L})^\top.$$

One may readily check that the gradient at x_0 is equal to

$$\nabla f(x_0) = (1, 0, \ldots, 0, 1)^\top$$

and that the minimum of the line search from x_0 in that direction is attained for step $\gamma = \frac{2}{L+\mu}$. One therefore obtains

$$x_1 = \left(\frac{L-\mu}{L+\mu}\right)(1/\mu, 0, \ldots, 0, -1/L)^\top,$$

and, for all $i = 0, 1, \ldots$

$$x_{2i} = \left(\frac{L-\mu}{L+\mu}\right)^{2i} x_0, \quad x_{2i+1} = \left(\frac{L-\mu}{L+\mu}\right)^{2i} x_1.$$

Since $f_* = 0$, it is straightforward to verify that equality

$$f(x_{i+1}) - f_* = \left(\frac{L-\mu}{L+\mu}\right)^2 (f(x_i) - f_*) \quad i = 0, 1, \ldots,$$

holds as required. □

The construction in Example 6.2 is illustrated in Figure 6.1 in the case $n = 2$, where the ellipses shown are level curves of the objective function. Each step from x_i to x_{i+1} is orthogonal to the ellipse at x_i (since it uses the steepest descent direction) and tangent to the ellipse at x_{i+1} (because of the exact line search direction), hence successive steps are orthogonal to each other.

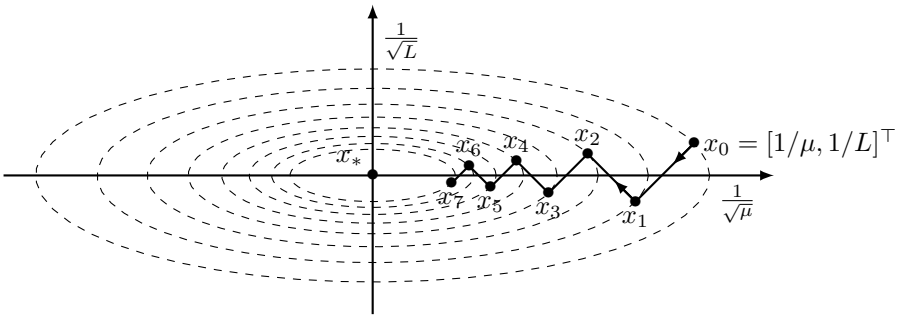


Figure 6.1: Illustration of Example 6.2 for the case $n = 2$ (small arrows indicate direction of negative gradient).

As an immediate consequence of Theorem 6.1 and Example 6.2, one has the following tight bound on the number of steps needed to obtain ϵ-relative accuracy on the objective function for a given $\epsilon > 0$.

Corollary 6.3. Given $\epsilon > 0$, the gradient method with exact line search yields a solution with relative accuracy ϵ for any function $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ after at most $N = \left\lceil \frac{1}{2} \log\left(\frac{1}{\epsilon}\right) / \log\left(\frac{L+\mu}{L-\mu}\right) \right\rceil$ iterations, i.e.

$$\frac{f(x_N) - f_*}{f(x_0) - f_*} \leq \epsilon,$$

where x_0 is the starting point. Moreover, this iteration bound is tight for the quadratic function defined in Example 6.2.

For non-quadratic functions in $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$, only bounds weaker than (6.1) are known. For example, in [LY08, p. 240], the following bound is shown:

$$(f(x_{i+1}) - f_*) \leq \left(1 - \frac{\mu}{L}\right)(f(x_i) - f_*) \quad i = 0, 1, \ldots$$

In [NW06, Theorem 3.4] a stronger result than Theorem 6.1 was claimed, but retracted in a subsequent erratum[1], which now only claims an asymptotic result.

A result related to Theorem 6.1 is given in [Nem99] where Armijo-rule line search is used instead of exact line search. An explicit rate in the strongly convex case is given there in Proposition 3.3.5 on page 53 (definition of the method is (3.1.2) on page 44). More general upper bounds on the convergence rates of gradient-type methods for convex functions may be found in the books [NY83, Nes04]. We mention one more particular result by Nesterov [Nes04] that is similar to our main result in Theorem 6.1, but that uses a fixed step size and relies on the initial distance to the solution.

Theorem 6.4 (Theorem 2.1.15 in [Nes04])**.** Given $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ and $x_0 \in \mathbb{R}^d$, the gradient descent method with fixed step length $\gamma = \frac{2}{\mu+L}$ generate iterates x_i $(i = 0, 1, 2, \ldots)$ that satisfy

$$f(x_i) - f_* \leq \frac{L}{2} \left(\frac{L-\mu}{L+\mu} \right)^{2i} \|x_0 - x_*\|^2 \quad i = 0, 1, \ldots$$

Note that this result does not imply Theorem 6.1.

6.2 Background results

In this section we collect some known results on strongly convex functions and on the gradient method. We will need these results in the proof of our main result, Theorem 6.1.

6.2.1 Properties of the gradient method with exact line search

Let x_i $(i = 1, 2, \ldots, N)$ be the iterates produced by the gradient method with exact line search started at x_0. Those iterates are defined by the following two conditions for $i = 0, 1, \ldots, N-1$

$$x_{i+1} - x_i + \gamma \nabla f(x_i) = 0, \text{ for some } \gamma \geq 0, \tag{6.2}$$

$$\langle \nabla f(x_{i+1}), x_{i+1} - x_i \rangle = 0 \tag{6.3}$$

where the first condition (6.2) states that we move in the direction of the negative gradient, and the second condition (6.3) expresses the exact line search condition.

[1] The erratum is available at: `http://users.iems.northwestern.edu/~nocedal/book/2ndprint.pdf`.

A consequence of those conditions is that successive gradients are orthogonal, i.e.,

$$\langle \nabla f(x_{i+1}), \nabla f(x_i) \rangle = 0 \quad i = 0, 1, \ldots, N-1. \tag{6.4}$$

Instead of relying on conditions (6.2)–(6.3) that define the iterates of the gradient method with exact line search, our analysis will be based on the weaker conditions (6.3)–(6.4), which are also satisfied by other sequences of iterates.

6.2.2 Performance estimation of the gradient method with exact line search

Consider the following SDP problem, for fixed parameters $N \geq 1$, $R > 0$, $\mu > 0$ and $L > \mu$:

$$\left.\begin{array}{lll} \max f_N - f_* & & \\ \text{s.t.} & & \\ \langle g_{i+1}, x_{i+1} - x_i \rangle & = & 0 \quad i \in \{0, 1, \ldots, N-1\} \\ \langle g_{i+1}, g_i \rangle & = & 0 \quad i \in \{0, 1, \ldots, N-1\} \\ \{(x_i, f_i, g_i)\}_{i \in \{*, 0, 1, \ldots, N\}} & \text{is} & \mathcal{F}_{\mu,L}\text{-interpolable} \\ g_* & = & 0 \\ f_0 - f_* & \leq & R, \end{array}\right\} \tag{6.5}$$

where the variables are $x_i \in \mathbb{R}^d$, $f_i \in \mathbb{R}^d$ and $g_i \in \mathbb{R}^d$ ($i \in \{*, 0, 1, \ldots, N\}$).

Lemma 6.5. The optimal value of the above SDP problem (6.5) is an upper bound on $f(x_N) - f_*$, where f is any function from $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$, f_* is its minimum and x_N is the Nth iterate of the gradient method with exact line search applied to f from any starting point x_0 that satisfies $f(x_0) - f_* \leq R$.

Proof. Fix any $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, and let $x_0, \ldots, x_N$ be the iterates of the gradient method with exact line search applied to f. Now a feasible solution to the SDP problem is given by

$$x_i,\ f_i = f(x_i),\ g_i = \nabla f(x_i) \quad i \in \{*, 0, \ldots, N\}.$$

The objective function value at this feasible point is $f_N = f(x_N)$, so that the optimal value of the SDP is an upper bound on $f(x_N) - f_*$. □

We are now ready to give a proof of our main result. We already mention that the SDP relaxation (6.5) is not used directly in the proof, but was used to devise the proof, in a sense that will be explained later.

6.3 Proof of Theorem 6.1

We only consider one iteration of the gradient method with exact line search, as we will see that it is sufficient to prove Theorem 6.1. Thus we consider only the first iterates, given by x_0 and x_1, as well as the minimizer x_* of $f \in \mathcal{F}_{\mu,L}$. Set $f_i = f(x_i)$ and $g_i = \nabla f(x_i)$ for $i \in \{*, 0, 1\}$. Note that $g_* = 0$. The following five inequalities are now satisfied:

$$
\begin{aligned}
1: \quad & f_0 \geq f_1 + \langle g_1, x_0 - x_1 \rangle + \frac{1}{2L}\|g_0 - g_1\|^2 \\
& \qquad + \frac{\mu}{2(1-\mu/L)}\left\|x_0 - x_1 - \frac{1}{L}(g_0 - g_1)\right\|^2, \\
2: \quad & f_* \geq f_0 + \langle g_0, (x_* - x_0)\rangle + \frac{1}{2L}\|g_0 - g_*\|^2 \\
& \qquad + \frac{\mu}{2(1-\mu/L)}\left\|x_0 - x_* - \frac{1}{L}(g_0 - g_*)\right\|^2, \\
3: \quad & f_* \geq f_1 + \langle g_1, (x_* - x_1)\rangle + \frac{1}{2L}\|g_1 - g_*\|^2 \\
& \qquad + \frac{\mu}{2(1-\mu/L)}\left\|x_1 - x_* - \frac{1}{L}(g_1 - g_*)\right\|^2, \\
4: \quad & 0 \geq \langle g_0, g_1 \rangle, \\
5: \quad & 0 \geq \langle g_1, x_1 - x_0 \rangle.
\end{aligned}
$$

Indeed, the first three inequalities are the $\mathcal{F}_{\mu,L}$-interpolability conditions, the fourth inequality is a relaxation of (6.4), and the fifth inequality is a relaxation of (6.3).

We aggregate these five inequalities by defining the following positive multipliers,

$$
y_1 = \frac{L-\mu}{L+\mu}, \quad y_2 = 2\mu\frac{(L-\mu)}{(L+\mu)^2}, \quad y_3 = \frac{2\mu}{L+\mu}, \quad y_4 = \frac{2}{L+\mu}, \quad y_5 = 1, \tag{6.6}
$$

and adding the five inequalities together after multiplying each one by the corresponding multiplier.

The result is the following inequality (as may be verified directly):

$$
\begin{aligned}
f_1 - f_* \quad \leq \quad & \left(\tfrac{L-\mu}{L+\mu}\right)^2 (f_0 - f_*) \\
& - \tfrac{\mu L(L+3\mu)}{2(L+\mu)^2}\left\|x_0 - \tfrac{L+\mu}{L+3\mu}x_1 - \tfrac{2\mu}{L+3\mu}x_* - \tfrac{3L+\mu}{L^2+3\mu L}g_0 - \tfrac{L+\mu}{L^2+3\mu L}g_1\right\|^2 \\
& - \tfrac{2L\mu^2}{L^2+2L\mu-3\mu^2}\left\|x_1 - x_* - \tfrac{(L-\mu)^2}{2\mu L(L+\mu)}g_0 - \tfrac{L+\mu}{2\mu L}g_1\right\|^2.
\end{aligned} \tag{6.7}
$$

Since the last two right-hand-side terms are nonpositive, we obtain:

$$f_1 - f_* \leq \left(\frac{L-\mu}{L+\mu}\right)^2 (f_0 - f_*).$$

Since x_0 was arbitrary, this completes the proof of Theorem 6.1. □

6.3.1 Remarks on the proof of Theorem 6.1.

- ◇ First, note that we have proven a bit more than what is stated in Theorem 6.1. Indeed, the result in Theorem 6.1 holds for any iterative method that satisfies the five inequalities used in its proof.
- ◇ Although the proof of Theorem 6.1 is easy to verify, it is not apparent how the multipliers $y_1, \ldots, y_5$ in (6.6) were obtained. This was in fact done via preliminary computations, and subsequently guessing the values in (6.6), through the following steps:
  1. The SDP performance estimation problem (6.5) with $N = 1$ was solved numerically for various values of the parameters μ , L and R — actually, the values of L and R can safely be fixed to some positive constants using appropriate scaling arguments (see e.g., Section 4.2.5 or [THG16a, Section 3.5] for a related discussion).
  2. The optimal values of the dual SDP multipliers of the constraints corresponding to the five inequalities in the proof gave the guesses for the correct values $y_1, \ldots, y_5$ as stated in in (6.6).
  3. Finally the correctness of the guess was verified directly (by symbolic computation and by hand).
- ◇ The key inequality (6.7) may be rewritten in another, more symmetric way

$$(f_1 - f_*) \leq (f_0 - f_*)\left(\frac{1-\kappa}{1+\kappa}\right)^2 - \frac{\mu}{4}\left(\frac{\|s_1\|^2}{1+\sqrt{\kappa}} + \frac{\|s_2\|^2}{1-\sqrt{\kappa}}\right),$$

where $\kappa = \mu/L$ is the condition number (between 0 and 1) and slack vectors s_1 and s_2 are

$$\begin{aligned} s_1 &= -\frac{(1+\sqrt{\kappa})^2}{1+\kappa}\left(x_0 - x_* - g_0/\sqrt{L\mu}\right) + \left(x_1 - x_* + g_1/\sqrt{L\mu}\right), \\ s_2 &= \frac{(1-\sqrt{\kappa})^2}{1+\kappa}\left(x_0 - x_* + g_0/\sqrt{L\mu}\right) - \left(x_1 - x_* - g_1/\sqrt{L\mu}\right). \end{aligned}$$

Note that the four expressions $x_i - x_* \pm g_i/\sqrt{L\mu}$ expressions are invariant

under dilation of f, and that cases of equality in (6.7) simply correspond to equalities $s_1 = s_2 = 0$.

◇ It is interesting to note that the known proof of Theorem 6.1 for the quadratic case only requires the so-called Kantorovich inequality, that may be stated as follows.

Theorem 6.6 (Kantorovich inequality; see e.g. Lemma 3.1 in [Ber99])**.** Let Q be a symmetric positive definite $n \times n$ matrix with smallest and largest eigenvalues $\mu > 0$ and $L \geq \mu$ respectively. Then, for any unit vector $x \in \mathbb{R}^d$, one has:

$$\left(x^\top Q x\right)\left(x^\top Q^{-1} x\right) \leq \frac{(\mu + L)^2}{4\mu L}.$$

Thus, the inequality (6.7) replaces the Kantorovich inequality in the proof of Theorem 6.1 for non-quadratic $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$.

◇ Finally, we note that this proof can easily be modified to handle the case of the fixed-step gradient method that was mentioned in Theorem 6.4. Indeed, observe that the proof aggregates the fourth and fifth inequalities with multipliers $y_4 = \frac{2}{L+\mu}$ and $y_5 = 1$, which leads to the combined inequality

$$\frac{-2}{L+\mu}\langle g_0, g_1\rangle + \langle g_1, x_0 - x_1\rangle \geq 0 \quad \Leftrightarrow \quad \langle g_1, x_0 - \frac{2}{L+\mu} g_0 - x_1\rangle \geq 0\ .$$

Now note that the gradient method with fixed step $\gamma = \frac{2}{L+\mu}$ satisfies this combined inequality (since the second factor in the left-hand side becomes zero), and hence the rest of the proof establishes the same rate for this method as for the gradient descent with exact line search.

Theorem 6.7. Let $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, x_* a global minimizer of f on $\mathbb{R}^d$, and $f_* = f(x_*)$. Each iteration of the gradient method with fixed step length $\gamma = \frac{2}{\mu+L}$ satisfies

$$f(x_{i+1}) - f_* \leq \left(\frac{L-\mu}{L+\mu}\right)^2 (f(x_i) - f_*) \quad i = 0, 1, \ldots$$

Note that Example 6.2 also establishes that this rate is tight. Hence we have the relatively surprising fact that, when looking at the worst-case convergence rate of the objective function accuracy, performing exact line search is not better than using a well-chosen fixed step length.

6.4 Extension to 'noisy' gradient descent with exact line search

Theorem 6.1 may be generalized to what we will call *noisy gradient descent method with exact linear search*; see e.g. [Ber99, p.59] where it is called gradient descent method with (relative) error. Here the search direction at iteration i, say d_i, satisfies

$$\| - \nabla f(x_i) - d_i\| \leq \varepsilon \|\nabla f(x_i)\| \quad i = 0, 1, \ldots, \tag{6.8}$$

where $0 \leq \varepsilon < 1$ is some given *relative* tolerance on the deviation from the negative gradient. Note that the algorithm cannot be guaranteed to converge as soon as $\varepsilon \geq 1$, since $d_i = 0$ then becomes feasible. We recover the normal gradient descent algorithm when $\varepsilon = 0$. Note that this model differs from the absolute inaccuracy on the subgradient that we considered on the proximal point algorithm (see Theorem 5.15).

In the case of more general values of ε, one can for example satisfy the relative error criterion by imposing a restriction of the type $|\sin\theta| \leq \varepsilon$ on the angle θ between search direction d_i and the current negative gradient $-\nabla f(x_i)$.

Using a search direction d_i that satisfies (6.8) corresponds, for example, to an implementation of the gradient descent method where each component of $-\nabla f(x_i)$ is only calculated to a fixed number of significant digits.

Thus we consider the following algorithm:

Noisy gradient descent method with exact line search

Input: $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, $x_0 \in \mathbb{R}^d$, $0 \leq \varepsilon < 1$.

for $i = 0, 1, \ldots$

Select any seach direction d_i that satisfies (6.8);

$\gamma = \operatorname{argmin}_{\gamma \in \mathbb{R}} f\left(x_i - \gamma d_i\right)$

$x_{i+1} = x_i - \gamma d_i$

One may show the following generalization of Theorem 6.1.

Theorem 6.8. Let $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, x_* a global minimizer of f on $\mathbb{R}^d$, and $f_* = f(x_*)$. Given a relative tolerance ε, each iteration of the noisy gradient descent method with exact line search satisfies

$$f(x_{i+1}) - f_* \leq \left(\frac{1 - \kappa_\varepsilon}{1 + \kappa_\varepsilon}\right)^2 (f(x_i) - f_*) \quad i = 0, 1, \ldots \tag{6.9}$$

where $\kappa_\varepsilon = \frac{\mu}{L}\frac{(1-\varepsilon)}{(1+\varepsilon)}$.

Proof. When $\varepsilon = 0$, the rate becomes $\frac{1-\kappa}{1+\kappa} = \frac{L-\mu}{L+\mu}$, which matches exactly Theorem 6.1, and the proof of Theorem 6.8 is a straightforward generalization of the proof of Theorem 6.1. The key is again to consider a wider class of iterative methods that satisfies certain inequalities. We use the following ones:

$$\begin{aligned}
1: \quad & f_0 \geq f_1 + \langle g_1, x_0 - x_1\rangle + \frac{1}{2L}\|g_0 - g_1\|^2 \\
& \qquad + \frac{\mu}{2(1-\mu/L)}\left\|x_0 - x_1 - \frac{1}{L}(g_0 - g_1)\right\|^2, \\
2: \quad & f_* \geq f_0 + \langle g_0, x_* - x_0\rangle + \frac{1}{2L}\|g_0 - g_*\|^2 \\
& \qquad + \frac{\mu}{2(1-\mu/L)}\left\|x_0 - x_* - \frac{1}{L}(g_0 - g_*)\right\|^2, \\
3: \quad & f_* \geq f_1 + \langle g_1, x_* - x_1\rangle + \frac{1}{2L}\|g_1 - g_*\|^2 \\
& \qquad + \frac{\mu}{2(1-\mu/L)}\left\|x_1 - x_* - \frac{1}{L}(g_1 - g_*)\right\|^2, \\
4: \quad & 0 \geq \langle g_1, x_1 - x_0\rangle, \\
5: \quad & 0 \geq \langle g_0, g_1\rangle - \varepsilon\|g_0\|\|g_1\|.
\end{aligned} \tag{6.10}$$

The first four inequalities are the same as before, and the fifth is satisfied by the iterates of the noisy gradient descent with exact line search. Indeed, in the first iteration one has:

$$\begin{aligned}
0 &= \frac{\langle d_0, g_1\rangle}{\|g_1\|} && \text{(exact line search)} \\
&= \frac{\langle d_0 + g_0, g_1\rangle}{\|g_1\|} - \frac{\langle g_0, g_1\rangle}{\|g_1\|} \\
&\leq \varepsilon\|g_0\| - \frac{\langle g_0, g_1\rangle}{\|g_1\|} && \text{(by Cauchy-Schwartz and (6.8)).}
\end{aligned}$$

We rewrite the fifth inequality as the equivalent linear matrix inequality:

$$\begin{pmatrix} \varepsilon\|g_0\|^2 & \langle g_0, g_1\rangle \\ \langle g_0, g_1\rangle & \varepsilon\|g_1\|^2 \end{pmatrix} \succeq 0. \tag{6.11}$$

We first aggregate the first four inequalities by adding them together after multiplication by the respective multipliers:

$$y_1 = \rho_\varepsilon, \quad y_2 = 2\kappa_\varepsilon\frac{1-\kappa_\varepsilon}{(1+\kappa_\varepsilon)^2}, \quad y_3 = \frac{2\kappa_\varepsilon}{1+\kappa_\varepsilon}, \quad y_4 = 1,$$

where $L_\varepsilon = (1+\varepsilon)L$, $\mu_\varepsilon = (1-\varepsilon)\mu$, $\kappa_\varepsilon = \frac{\mu_\varepsilon}{L_\varepsilon}$ and $\rho_\varepsilon = \frac{1-\kappa_\varepsilon}{1+\kappa_\varepsilon}$.

Next we define a positive semidefinite matrix multiplier for the linear matrix inequality (6.11), namely

$$\begin{pmatrix} a\rho_\varepsilon & -a \\ -a & \frac{a}{\rho_\varepsilon} \end{pmatrix} \succeq 0, \tag{6.12}$$

with $a = \frac{1}{L_\varepsilon + \mu_\varepsilon}$, and add nonnegativity of the inner product between the left-hand-side of (6.11) and the multiplier matrix (6.12) to the aggregated constraints. It can now be checked that the resulting expression is the following generalization of (6.7)

$$\begin{aligned} f_1 - f_* \leq & \rho_\varepsilon^2 (f_0 - f_*) \\ & - \frac{L\mu(L_\varepsilon - \mu_\varepsilon)(L_\varepsilon + 3\mu_\varepsilon)}{2(L-\mu)(L_\varepsilon + \mu_\varepsilon)^2} \|x_0 + \alpha_1 x_1 - (1+\alpha_1)x_* + \alpha_2 g_0 + \alpha_3 g_1\|^2 \\ & - \frac{2L\mu\mu_\varepsilon}{(L-\mu)(L_\varepsilon + 3\mu_\varepsilon)} \|x_1 - x_* + \alpha_4 g_0 + \alpha_5 g_1\|^2, \end{aligned}$$

with the appropriate coefficients

$$\begin{gathered} \alpha_1 = -\frac{L_\varepsilon + \mu_\varepsilon}{L_\varepsilon + 3\mu_\varepsilon},\ \alpha_2 = -\frac{4L - L_\varepsilon + \mu_\varepsilon}{L(L_\varepsilon + 3\mu_\varepsilon)},\ \alpha_3 = \frac{(L_\varepsilon + \mu_\varepsilon)(-4L + 3L_\varepsilon + \mu_\varepsilon)}{L(L_\varepsilon - \mu_\varepsilon)(L_\varepsilon + 3\mu_\varepsilon)}, \\ \alpha_4 = -\frac{(L-\mu)(L_\varepsilon - \mu_\varepsilon)}{2L\mu(L_\varepsilon + \mu_\varepsilon)},\ \alpha_5 = -\frac{L+\mu}{2L\mu}. \end{gathered}$$

This completes the proof. □

To conclude this section, the following example, based on the same quadratic function as Example 6.2, shows that our bound (6.9) for the noisy gradient descent is also tight.

Example 6.9. Consider the same quadratic function as in Example 6.2:

$$f(x) = \frac{1}{2} x^\top \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_d)\ x, \quad \text{where} \quad 0 < \mu = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_d = L.$$

Let θ be an angle satisfying $0 \leq \theta < \frac{\pi}{2}$. Consider the noisy gradient descent method where direction d_0 is obtained by performing a counter-clockwise 2D-rotation with angle θ on the first and last coordinates of the gradient $\nabla f(x_0)$. As mentioned above, this satisfies our definition with relative tolerance $\varepsilon = \sin\theta$. Define now the starting point

$$x_0 = \left(\frac{1}{\mu}, 0, \ldots, 0, \frac{1}{L}\sqrt{\frac{1-\varepsilon}{1+\varepsilon}} \right)^\top.$$

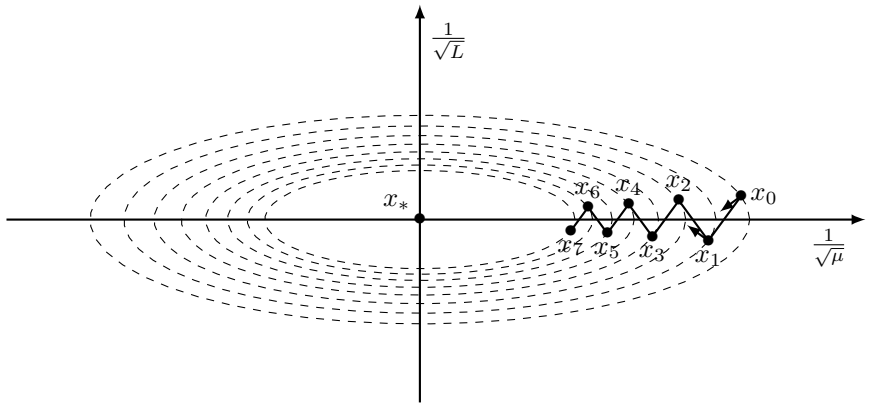


Figure 6.2: Illustration Example 6.9 for $n = 2$ and $\varepsilon = 0.3$ (small arrows indicate direction of negative gradient).

Tedious but straightforward computations show that

$$x_1 = \left(\frac{1-\kappa_\varepsilon}{1+\kappa_\varepsilon}\right)\left(\frac{1}{\mu}, 0, \ldots, 0, -\frac{1}{L}\sqrt{\frac{1-\varepsilon}{1+\varepsilon}}\right)^\top \quad \text{where } \kappa_\varepsilon = \frac{\mu}{L}\frac{(1-\varepsilon)}{(1+\varepsilon)}.$$

Moreover, if one chooses d_1 by rotating the second gradient $\nabla f(x_1)$ by the same angle θ in the clockwise direction, one obtains

$$x_2 = \left(\frac{1-\kappa_\varepsilon}{1+\kappa_\varepsilon}\right)^2\left(\frac{1}{\mu}, 0, \ldots, 0, \frac{1}{L}\sqrt{\frac{1-\varepsilon}{1+\varepsilon}}\right)^\top = \left(\frac{1-\kappa_\varepsilon}{1+\kappa_\varepsilon}\right)^2 x_0.$$

A similar reasoning for the next iterates, alternating clockwise and counter-clockwise rotations, shows that

$$x_{2i} = \left(\frac{1-\kappa_\varepsilon}{1+\kappa_\varepsilon}\right)^{2i} x_0, \quad x_{2i+1} = \left(\frac{1-\kappa_\varepsilon}{1+\kappa_\varepsilon}\right)^{2i} x_1 \text{ for all } i = 0, 1, \ldots$$

and hence we have that equality

$$f(x_{i+1}) - f_* = \left(\frac{1-\kappa_\varepsilon}{1+\kappa_\varepsilon}\right)^2 (f(x_i) - f_*) \quad i = 0, 1, \ldots$$

holds as announced. Figure 6.2 displays a few iterates, and can be compared to Figure 6.1. □

Before concluding the chapter, let us formulate a more practical version of Theorem 6.8 (which also applies for Theorem 6.1).

Corollary 6.10. Let $f \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$, $Q \subset \mathbb{R}^d$ be the sublevel set $Q = \{x : f(x) \leq f(x_0)\}$, and $x_* \in Q$ be optimal. Assuming that $\mu I_d \preceq \nabla^2 f(x) \preceq L I_d$ for any $x \in Q$, each iteration of the noisy gradient method with exact line search satisfies

$$f(x_{i+1}) - f_* \leq \left(\frac{1-\kappa_\varepsilon}{1+\kappa_\varepsilon}\right)^2 (f(x_i) - f_*) \quad i = 0, 1, \ldots \tag{6.13}$$

where $\kappa_\varepsilon = \frac{\mu}{L}\frac{(1-\varepsilon)}{(1+\varepsilon)}$.

Proof. First, note that $\mu I_d \preceq \nabla^2 f(x) \preceq L I_d$ for any $x \in Q$ implies that f locally satisfies Conditions 2.4 (smoothness) and 2.7 (strong convexity).

Hence, one can apply Corollary 2.60 in order to obtain an extended function $\tilde{f} \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ such that $f(x) = \tilde{f}(x)$ for all $x \in Q$ (which is closed by continuity of f, and convex by convexity of f).

Then, it suffices to apply Theorem 6.1 to $\tilde{f}(x)$, which is equal to f for all iterates (because each iteration improve the value of the objective function, and hence each new iterate belongs to the original sublevel set Q). □

6.5 Conclusion

The main results of this chapter are the exact convergence rates of the gradient descent method with exact line search and its noisy variant for strongly convex functions with Lipschitz continuous gradients. The computer-assisted technique of proof is also of independent interest, and demonstrates the importance of the SDP performance estimation problems (PEPs) introduced in [DT14].

Indeed, to obtain our proof of Theorem 6.8, the following SDP PEP was solved numerically for various fixed values of R, μ and L:

$$\max f_1 - f_* \text{ subject to (6.10) and } f_0 - f_* \leq R.$$

It was observed that, for each set of values, the optimal value of the SDP corresponded exactly to the bound in Theorem 6.8 (actually, for homogeneity reasons, L and R could be fixed and only μ needed to vary). Based on this, a rigorous proof Theorem 6.8 could be given by guessing the correct values of the dual SDP multipliers as functions of μ, L and R, and then verifying the guess through an explicit computation.

We believe this type of computer-assisted proof could prove useful in the analysis of more methods where exact line search is used (see for example Section 5.3.5 where we studied a fixed step conditional gradient method; could we use line search instead ?).

PEPs have been used by now to study worst-case convergence rates of several first-order optimization methods (see Chapters 4, 5 and [DT14, THG16a, THG16b]). This work differs in an important aspect: the performance estimation problem considered actually characterizes a whole class of methods that contains the method of interest (gradient descent with exact line search) as well as many other methods. This relaxation in principle only provides an upper bound on the worst-case of gradient descent, and it is the fact that Example 6.2 matches this bound that allows us to conclude with a tight result.

The reason we could not solve the performance estimation problem for the gradient descent method itself is that equation (6.2), which essentially states that the step $x_{i+1} - x_i$ is parallel to the gradient $\nabla f(x_i)$, cannot be formulated as a convex constraint in the SDP formulation. The main obstruction appears to be that requiring that two vectors are parallel is a nonconvex constraint, even when working with their inner products[2]. Instead, our convex formulation enforces that those two vectors are both orthogonal to a third one, the next gradient $\nabla f(x_{i+1})$.

[2]One such nonconvex formulation would be $\langle g_i, x_i - x_{i+1} \rangle = \|g_i\| \|x_i - x_{i+1}\|$.

Chapter 7

Proximal Gradient Method

In this chapter, we establish tight convergence rates for the proximal gradient method applied to the sum of a smooth strongly convex function and a non-smooth convex function with a proximal operator available. Those convergence guarantees are shown to be valid for different standard performance measures: objective function accuracy, distance to optimality and residual (sub)gradient norm. The global and exact worst-case guarantees we present for the proximal gradient method are conceptually very simple, although apparently new.

On the way, our results allow explicitly weakening the assumptions for obtaining the corresponding linear convergence guarantees, establish that the fixed step size policy $\frac{2}{L+\mu}$ is optimal for decreasing the distance to optimality, objective function accuracy and residual gradient norm, and extend a recent result of Chapter 6 (see also [dKGT16]) on the worst-case behavior of steepest descent with exact line search to the non-smooth convex composite case.

This chapter is divided into three main parts:

- ◇ Section 7.1 presents the context, some particular cases and known convergence results for the proximal gradient method in the strongly convex case.
- ◇ In Section 7.2, we provide simple lower bounds along with matching upper bound for standard convergence measures: objective function accuracy (OFA), distance to optimality (DO) and residual (sub)gradient norm (RGN).
- ◇ In Section 7.3, we present convergence results for mixed initial and final convergence measures (DO, RGN and OFA).
- ◇ Finally, we conclude and propose further research directions in Section 7.4.

The subsequent text is based on work [THG16c].

In the following, we work in the Euclidean space $\mathbb{R}^d$ endowed with the inner product $\langle .,. \rangle : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ and the corresponding Euclidean norm $\|x\|^2 = \langle x, x \rangle \ \forall x \in \mathbb{R}^d$. Nevertheless, the analyses are also directly valid in any finite dimensional real vector space $\mathbb{E}$ and the corresponding dual space $\mathbb{E}^*$ with Euclidean structures, as introduced in Section 2.1.

7.1 Introduction

We consider the two-term composite strongly convex optimization setting (instance of the general composite optimization problem (CM))

$$\min_{x \in \mathbb{R}^d} \{F(x) \equiv f(x) + h(x)\} \tag{7.1}$$

where $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ is a L-smooth μ-strongly convex proper function over $\mathbb{R}^d$, for some $0 < \mu \leq L$ and $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ is convex, closed and proper over $\mathbb{R}^d$ (i.e., we only consider strongly convex functions). In addition, we assume that we can evaluate the gradient of f and the proximal operator of h [PB13, Section 1.1]:

$$p_{\gamma h}(x) = \underset{y \in \mathbb{R}^d}{\operatorname{argmin}} \left\{ \gamma h(y) + \frac{1}{2}\|x - y\|^2 \right\}. \tag{PROX}$$

Also, we use the proximal gradient method (PGM) with constant step length γ to solve (7.1).

Proximal gradient method (PGM)

Input: $x_0 \in \mathbb{R}^d$, $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$, $0 \leq \gamma \leq \frac{2}{L}$.

For $k = 0 : N-1$

$$x_{k+1} = p_{\gamma h}(x_k - \gamma \nabla f(x_k))$$

For notational convenience, we denote by $s_{k+1} \in \partial h(x_{k+1})$ the subgradient of h used in the proximal operation (more details in Section 7.2.2), that is

$$s_{k+1} = \frac{x_k - x_{k+1}}{\gamma} - \nabla f(x_k). \tag{7.2}$$

Note that $\nabla f(x_{k+1}) + s_{k+1} \in \partial F(x_{k+1})$ is a particular subgradient of F at x_{k+1} that we can actually compute using (7.2) and whose convergence is studied hereafter. In the sequel, we will often use the compact notation $\tilde{\nabla} F(x)$ to denote a subgradient of F at x; hence $\tilde{\nabla} F(x) \in \partial F(x)$.

Example 7.1. The composite convex problem (7.1) has the following very common particular cases:

- the unconstrained minimization problem $\min_{x\in\mathbb{R}^d} f(x)$ when $h(x) = 0$ and $p_{\gamma h}(x) = x$. In this case, PGM is simply the standard unconstrained gradient method (UGM) $x_{k+1} = x_k - \gamma\nabla f(x_k)$.
- The constrained minimization problem $\min_{x\in Q} f(x)$ with $Q \subseteq \mathbb{R}^d$ a closed convex set. This corresponds to choosing $h(x) = i_Q(x)$ (i_Q is the indicator function of Q) for which the proximal operation corresponds to a projection onto Q: $p_{\gamma h}(x) = \Pi_Q(x)$. In this case, PGM is simply the standard projected gradient method (ΠGM) $x_{k+1} = \Pi_Q(x_k - \gamma\nabla f(x_k))$.
- The composite minimization problem $\min_{x\in\mathbb{R}^d} f(x) + h(x)$ where $h(x)$ has an analytical proximal operator available[1] (e.g., the classical ℓ_1-regularization term $h(x) = \|x\|_1$).

Convergence rate. In the sequel, we use the notation (valid for $0 \leq \gamma \leq \frac{2}{L}$, $0 < \mu \leq L < \infty$)

$$\rho(\gamma) = \max\{|1 - L\gamma|, |1 - \mu\gamma|\} = \max\{(L\gamma - 1), (1 - \mu\gamma)\}, \qquad \text{(RHO)}$$

so that $\rho(\gamma) \geq 0$ for all values of the step size γ such that $0 \leq \gamma \leq \frac{2}{L}$ and $0 < \mu \leq L < \infty$. We prove in Section 7.2.3 that applying PGM to Problem (7.1) produces a sequence of iterates converging with rate $\rho^2(\gamma)$ in distance to optimality, residual gradient norm and objective function accuracy.

Note that the term $(1 - L\gamma)^2$ in the expression of $\rho^2(\gamma)$ is minimized by taking the so-called *short step* $1/L$, whereas the second term $(1 - \mu\gamma)^2$ is minimized by choosing the so-called *long step* $1/\mu$. A direct implication is that the best possible worst-case convergence rate is achieved by the step size $\frac{2}{L+\mu}$ for all three performance measures (this is illustrated on Figure 7.1).

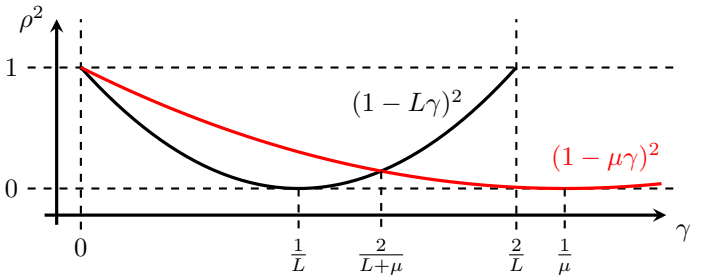


Figure 7.1: Rate of convergence $\rho^2(\gamma)$ as a function of the step size γ.

Our main result can now be stated.

[1] A list of useful analytical proximal operators is available in [CP11].

Theorem 7.2. Let $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, and $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ and consider the composite convex optimization problem (7.1) and a feasible starting point $x_0 \in \mathbb{R}^d$ (i.e., x_0 is such that $F(x_0) < \infty$). The iterates of PGM with $0 \leq \gamma \leq \frac{2}{L}$ satisfy the following $\forall k = 0, 1, 2, \ldots$:

$$\max_{\substack{f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d) \\ h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d) \\ x_0 \in \mathbb{R}^d}} \left\{ \frac{\|x_k - x_*\|^2}{\|x_0 - x_*\|^2} \right\} = \rho^{2k}(\gamma),$$

$$\max_{\substack{f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d) \\ h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d) \\ x_0 \in \mathbb{R}^d \\ s_0 \in \partial h(x_0)}} \left\{ \frac{\|\nabla f(x_k) + s_k\|^2}{\|\nabla f(x_0) + s_0\|^2} \right\} = \rho^{2k}(\gamma),$$

$$\max_{\substack{f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d) \\ h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d) \\ x_0 \in \mathbb{R}^d}} \left\{ \frac{F(x_k) - F(x_*)}{F(x_0) - F(x_*)} \right\} = \rho^{2k}(\gamma),$$

where $x_* \in \mathbb{R}^d$ denotes the optimal solution of (7.1), and s_k denotes the subgradient used in the proximal operation to generate x_k (see Equation 7.2).

Prior works. The UGM and ΠGM are standard methods whose analysis in the context of smooth strongly convex functions can be found in numerous references. The convergence in distance to optimality according to

$$\|x_{k+1} - x_*\|^2 \leq \rho^2(\gamma)\|x_k - x_*\|^2, \tag{7.3}$$

can be found in e.g., [Pol87, Section 1.4: Theorem 3], [RB16, Section 5.1] and [LRP16, Section 4.4] for UGM and ΠGM, with slight variations in the assumptions (depending on whether or not f is required to be twice differentiable). For the specific step size $1/L$, the guarantee (7.3) can be found as a particular case of [SLRB11, Proposition 3]. Also, weaker convergence rates such as $(1 - \frac{\mu}{L})$ for the specific step size $1/L$ can be found in e.g., [Nes04, Theorem 2.2.8] or [Bub15, Theorem 3.10] for ΠGM. Also note that (7.3) also holds PGM, as it essentially follows the same proof technique as for ΠGM (using the non-expansiveness of the proximal operation).

As far as we know, results in terms of $F(x_N) - F_*$ or $\left\|\tilde{\nabla} F(x_N)\right\|$ are typically not as emphasized (or known) as compared to convergence in terms of $\|x_N - x_*\|$. However, it is standard to convert results in terms of $\|x_N - x_*\|$ to $\left\|\tilde{\nabla} F(x_N)\right\|$

and $F(x_N) - F_*$ using the smoothness and strong convexity assumptions. In the particular case of unconstrained minimization (i.e., $h = 0$), one can use:

$$f(x_k) - f(x_*) \leq \frac{L}{2}\|x_k - x_*\|^2, \ \|\nabla f(x_k)\| \leq L\|x_k - x_*\|,$$
$$f(x_k) - f(x_*) \geq \frac{\mu}{2}\|x_k - x_*\|^2, \ \|\nabla f(x_k)\| \geq \mu\|x_k - x_*\|,$$

in order to adapt the convergence in terms of distance to optimality to convergence in objective function accuracy and residual gradient norm:

$$f(x_k) - f(x_*) \leq \frac{L}{\mu}\rho^{2k}(\gamma)(f(x_0) - f(x_*)), \text{ and } \|\nabla f(x_k)\| \leq \frac{L}{\mu}\rho^{k}(\gamma)\|\nabla f(x_0)\|. \tag{7.4}$$

The bounds (7.4) are not tight because of the leading constant L/μ (see Theorem 7.2). In addition, we typically have $\frac{L}{\mu}\rho^{2k} > 1$ for small values of k, and therefore the former inequality (7.4) does not even guarantee an improvement in terms of objective function accuracy or residual gradient norm for few iterations.

The global convergence rate $\rho^2\left(\frac{2}{L+\mu}\right) = \left(\frac{L-\mu}{L+\mu}\right)^2$ in terms of objective function accuracy was only very recently obtained for UGM with step size $\frac{2}{L+\mu}$ as a by-product of the convergence guarantee of using exact line search for solving unconstrained smooth convex minimization problems [dKGT16, Theorem 1.2], whereas previous results were establishing a $\left(1 - \frac{\mu}{L}\right)$ global convergence rate (see e.g., [Ber99, LY08]). Theorem 7.2 further extends this result in the composite case (7.1) for the different convergence measures, for embedding a projection or a proximal step and for all reasonable step sizes. As a by-product, we generalize Theorem 6.1 (see also [dKGT16, Theorem 1.2]) to proximal gradient methods with line search (see Section 7.4).

7.2 Convergence in distance, gradient and function accuracy

7.2.1 Quadratic lower bounds

First, we focus on the case of a quadratic function f without any nonsmooth term ($h = 0$), which provides us with lower complexity bounds for the different values of the step size γ. Those quadratics correspond to lower bounds for UGM and therefore also for ΠGM and PGM. We will show that those are tight for the class of smooth strongly convex functions in the following section.

Consider two constants $0 < \mu \leq L < +\infty$ and the corresponding quadratic functions $f_\mu(x) = \frac{\mu}{2}\|x\|^2$ and $f_L(x) = \frac{L}{2}\|x\|^2$. We clearly have that $f_\mu, f_L \in$

$\mathcal{F}_{\mu,L}(\mathbb{R}^d)$ and that $x_* = 0$ and $f_* = 0$ for both functions. In addition, one iteration of UGM on those functions respectively gives:

$$x_{k+1}^{(\mu)} = (1-\gamma\mu)x_k^{(\mu)}, \quad x_{k+1}^{(L)} = (1-\gamma L)x_k^{(L)},$$

which respectively lead to

$$\begin{aligned} f_\mu(x_{k+1}^{(\mu)}) &= \tfrac{\mu}{2}(1-\gamma\mu)^2\left\|x_k^{(\mu)}\right\|^2, \quad \left\|\nabla f_\mu(x_{k+1}^{(\mu)})\right\|^2 = (1-\gamma\mu)^2\left\|\nabla f_\mu(x_k^{(\mu)})\right\|^2, \\ f_L(x_{k+1}^{(L)}) &= \tfrac{L}{2}(1-\gamma L)^2\left\|x_k^{(L)}\right\|^2, \quad \left\|\nabla f_L(x_{k+1}^{(L)})\right\|^2 = (1-\gamma L)^2\left\|\nabla f_L(x_k^{(L)})\right\|^2. \end{aligned}$$

Those equalities allow to conclude that the worst-case behaviour for any of the criterion $f(x_{k+1}) - f_*$, $\|x_{k+1} - x_*\|^2$ and $\|\nabla f(x_{k+1})\|^2$ is at least as bad as in the cases of those two functions. That is, for any $\gamma \in \mathbb{R}$, there exists a $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ such that one iteration of UGM gives for all $k \geq 0$:

$$\begin{aligned} \|x_k - x_*\|^2 &= \rho^{2k}(\gamma)\|x_0 - x_*\|^2, \\ f(x_k) - f_* &= \rho^{2k}(\gamma)(f(x_0) - f_*), \\ \|\nabla f(x_k)\|^2 &= \rho^{2k}(\gamma)\|\nabla f(x_0)\|^2. \end{aligned} \tag{QLB}$$

We will see that that no other function behaves (strictly) worse.

7.2.2 Basic inequalities characterizing one iteration of the proximal gradient method

Now, we make a short inventory of the inequalities available to prove the different global convergence rates. Note that recent works on performance estimation of first-order methods (see [THG16a, THG16b]) guarantee that no other inequalities are needed in order to obtain the desired convergence results.

In the following, we denote by g_i and s_i the (sub)gradients of respectively the smooth function f and the non-smooth component h at the iteration k; that is $g_k = \nabla f(x_k)$ and $s_k \in \partial h(x_k)$, and by f_k and h_k the function values at those points: $f_k = f(x_k)$ and $h_k = h(x_k)$. In addition to that, we denote by x_* the optimal point (unique by strong convexity of F) and by $g_* = \nabla f(x_*)$ and $s_* \in \partial h(x_*)$ the gradient and some subgradient of respectively f and h at the optimum. Let us list the (in)equalities that enables us to characterize one iteration of PGM.

(a) The iteration $x_{k+1} = p_{\gamma h}\left(x_k - \gamma\nabla f(x_k)\right)$ can be rewritten using necessary and sufficient optimality conditions on the definition of the proximal operation (PROX):

$$x_{k+1} = x_k - \gamma(g_k + s_{k+1})$$

for some $s_{k+1} \in \partial h(x_{k+1})$.

(b) Optimality of x_* for (7.1) amounts to requiring $g_* + s_* = 0$ for some $s_* \in \partial h(x_*)$.

(c) For characterizing smoothness and strong convexity, we use the conditions from [THG16a, Theorem 4]. This should be required between three points: x_k, x_{k+1} and x_*. That is, $\forall i \neq j \in \{k, k+1, *\}$ (i.e., for the six possible pairs (i, j) within $\{k, k+1, *\}$) we have:

$$\begin{aligned} f_i \geq f_j \quad & + \langle g_j, x_i - x_j \rangle + \tfrac{1}{2L} \|g_i - g_j\|^2 \\ & + \tfrac{\mu}{2(1-\mu/L)} \left\| x_i - x_j - \tfrac{1}{L}(g_i - g_j) \right\|^2. \end{aligned} \tag{7.5}$$

(d) Similarly, we require that for all pairs (i, j): $i \neq j \in \{k, k+1, *\}$ (again, for the six possible combinations):

$$h_i - h_j - \langle s_j, x_i - x_j \rangle \geq 0, \tag{7.6}$$

for characterizing the (possibly non-smooth) convex function h.

7.2.3 Tight upper bounds

In this section, we prove the main convergence results of the paper, beginning with the convergence in terms of distance to optimality.

Distance to optimality. As provided in Section 7.1, the following convergence result in term of distance to optimality is not new. For the sake of clarity and completeness, we begin by proving it using the same technique that will be used for the subsequent results (residual gradient norm and objective function accuracy). The proof methodology relies from the performance estimation methodology (see [DT14, THG16a, THG16b, Dro14]). This technique has the advantage of being transparent and of explicitly showing *minimal assumptions* for obtaining this convergence property (see discussion below)..

Theorem 7.3 (Distance to optimality)**.** Consider the composite convex optimization problem (7.1). Every pair of consecutive iterates of the PGM with $0 \leq \gamma \leq \frac{2}{L}$ satisfies the following inequality:

$$\|x_{k+1} - x_*\|^2 \leq \rho^2(\gamma) \|x_k - x_*\|^2.$$

Proof. We use the notations and inequalities introduced in the previous section (Section 7.2.2) in order to construct the proof. As proposed in Section 7.2.2, we use some of the interpolation inequalities (7.5) and (7.6) between the iterates

and the optimal point. The proof consists in summing those interpolation inequalities after multiplying them with their respective coefficients (multipliers λ's).

First, we use (7.5) with respectively $(i,j) = (*,k)$ and $(i,j) = (k,*)$:

$$\begin{aligned} f_* \geq f_k \quad & +\langle g_k, x_* - x_k\rangle + \tfrac{1}{2L}\|g_k - g_*\|^2 \\ & +\tfrac{\mu}{2(1-\mu/L)}\left\|x_k - x_* - \tfrac{1}{L}(g_k - g_*)\right\|^2 \qquad : \lambda_0, \\ f_k \geq f_* \quad & +\langle g_*, x_k - x_*\rangle + \tfrac{1}{2L}\|g_k - g_*\|^2 \\ & +\tfrac{\mu}{2(1-\mu/L)}\left\|x_k - x_* - \tfrac{1}{L}(g_k - g_*)\right\|^2 \qquad : \lambda_1 \end{aligned}$$

Then, we use (7.6) with respectively $(i,j) = (*,k+1)$ and $(i,j) = (k+1,*)$:

$$\begin{aligned} h_* &\geq h_{k+1} + \langle s_{k+1}, x_* - x_{k+1}\rangle \qquad : \lambda_2, \\ h_{k+1} &\geq h_* + \langle s_*, x_{k+1} - x_*\rangle \qquad : \lambda_3. \end{aligned}$$

We use the following multipliers

$$\lambda_0 = \lambda_1 = 2\gamma\rho(\gamma) \geq 0, \quad \lambda_2 = \lambda_3 = 2\gamma \geq 0.$$

After appropriate substitutions of x_{k+1} and s_*, using $x_{k+1} = x_k - \gamma(g_k + s_{k+1})$ (Section 7.2.2, Condition (a)) and $s_* = -g_*$ (Section 7.2.2, Condition (b)) , and with little effort, one can check that the previous weighted sum of inequalities can be written in one of the following forms. We divide the proof in two cases (corresponding to the two regimes of $\rho(\gamma)$, see (RHO)).

- ◇ When $0 \leq \gamma \leq \frac{2}{L+\mu}$ (i.e., $\rho(\gamma) = (1-\gamma\mu)$), the expression can be written as

$$\begin{aligned} (1-\gamma\mu)^2\|x_k - x_*\|^2 \geq & \|x_{k+1} - x_*\|^2 + \gamma^2\|g_* + s_{k+1}\|^2 \\ & + \frac{\gamma(2-\gamma(L+\mu))}{L-\mu}\|\mu(x_k - x_*) - g_k + g_*\|^2, \\ \geq & \|x_{k+1} - x_*\|^2, \end{aligned}$$

  where the last inequality follows from

$$\gamma^2 \geq 0, \ \gamma(2-\gamma(L+\mu)) \geq 0, \text{ and } L-\mu \geq 0.$$

- ◇ Similarly, when $\frac{2}{L+\mu} \leq \gamma \leq \frac{2}{L}$ (i.e., $\rho(\gamma) = (L\gamma - 1)$), the expression is

equivalent to

$$\begin{aligned}(1-\gamma L)^2\left\|x_k-x_*\right\|^2 \geq & \left\|x_{k+1}-x_*\right\|^2+\gamma^2\left\|g_*+s_{k+1}\right\|^2 \\ & +\frac{\gamma(\gamma(L+\mu)-2)}{L-\mu}\left\|L(x_k-x_*)-g_k+g_*\right\|^2, \\ \geq & \left\|x_{k+1}-x_*\right\|^2,\end{aligned}$$

where the last inequality follows from

$$\gamma^2 \geq 0,\ \gamma(\gamma(L+\mu)-2) \geq 0,\ \text{and}\ L-\mu \geq 0.$$

We note that for any γ such that $0 \leq \gamma \leq \frac{2}{L}$, exactly[2] one of the two previous combinations of inequalities is valid (both multipliers and coefficients of the squared norms are positive). In addition, the valid expression corresponds to the maximum value between the two possible rates $(1-\gamma\mu)^2$ and $(1-\gamma L)^2$, which concludes the proof. □

Corollary 7.4. Let $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, and $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ and consider the composite convex optimization problem (7.1). The iterates of PGM with $0 \leq \gamma \leq \frac{2}{L}$ satisfy the following:

$$\max_{\substack{f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d) \\ h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d) \\ x_0 \in \mathbb{R}^d}} \left\{ \frac{\left\|x_k-x_*\right\|^2}{\left\|x_0-x_*\right\|^2} \right\} = \rho^{2k}(\gamma).$$

Proof. Combine Theorem 7.3 with the quadratic lower bounds (QLB). □

Before moving to the next convergence result, note that only a subset of the available inequalities were used in the previous proof. In fact, any composite function F for which the f component satisfies $\forall x \in \mathbb{R}^d$:

$$\langle g_*-g_k, x_*-x_k\rangle \geq \frac{1}{L}\left\|g_k-g_*\right\|^2+\frac{\mu}{1-\mu/L}\left\|x_k-x_*-\frac{1}{L}(g_k-g_*)\right\|^2, \quad (7.7)$$

(which is, sum of the two first inequalities used in the proof of Theorem 7.3, as $\lambda_0=\lambda_1$) will have a PGM converging with the same rate in terms of distance to optimality despite being potentially outside of $\mathcal{F}_{\mu,L}$. As an example, consider the following quadratic function $f_A(x)=\frac{1}{2}x^\top Ax$ with $\mu I \preceq A \preceq LI$ (hence

[2] Actually, both regimes are valid for $\gamma=\frac{2}{L+\mu}$.

$f \in \mathcal{F}_{\mu,L}$). Therefore, (7.7) holds and hence:

$$\left(1+\frac{\mu}{L}\right)(x_*-x_k)^\top A(x_*-x_k) \geq \frac{1}{L}(x_k-x_*)^\top A^\top A(x_k-x_*) + \mu(x_k-x_*)^\top(x_k-x_*).$$

In short, note that this inequality also holds when instead $0 \preceq A \preceq LI$ where x_* is the projection of x_k onto the set of optimal solutions (i.e., $x_k - x_* \perp \text{Null}(A)$) and $\mu > 0$ is the smallest nonzero eigenvalue of A.

Also note that only a monotonicity condition on ∂h needs to be satisfied for keeping the same convergence guarantees, as only the sum of the third and fourth inequalities is required to hold ($\lambda_2 = \lambda_3$):

$$\langle s_{k+1} - s_*, x_{k+1} - x_* \rangle \geq 0,$$

Note that those sorts of relaxations were further exploited in [ZC15, NNG15] (relaxation of the strong convexity requirement, with motivational examples). We leave further investigations in that direction for future research.

Residual gradient norm. The next theorem is concerned with convergence in terms of residual gradient norm. Note that similar results can be obtained for the norm of the (composite) gradient mapping (i.e., $\frac{x_k - x_{k+1}}{\gamma}$) instead[3], which is used in some standard references on composite minimization [Nes04, Nes13].

Convergence in terms of residual gradient norm is in fact very natural, as it is measurable in practice, as opposed to the distance to optimality which requires the knowledge of x_* in order to be evaluated, or in terms of objective function accuracy which it requires the knowledge (or a least a bound) on the true value of $F(x_*)$.

Theorem 7.5 (Residual gradient norm)**.** Let $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, and $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ and consider the composite convex optimization problem (7.1) and a feasible starting point $x_0 \in \mathbb{R}^d$ (i.e., x_0 is such that $F(x_0) < \infty$) such that there exists $s_0 \in \partial h(x_0)$. The iterates of PGM with $0 \leq \gamma \leq \frac{2}{L}$ satisfy:

$$\|\nabla f(x_{k+1}) + s_{k+1}\|^2 \leq \rho^2(\gamma)\|\nabla f(x_k) + s_k\|^2,$$

with $s_k \in \partial h(x_k)$ (any subgradient of h at x_k) and $s_{k+1} \in \partial h(x_{k+1})$, the subgradient of h at x_{k+1} used in the proximal operation (see Equation (7.2)).

Proof. We use the exact same reasoning as for Theorem 7.3: the notations and inequalities are introduced in the previous section (Section 7.2.2), and

[3]The difference between the gradient mapping and the residual gradient norm is simple, but somewhat subtle. The gradient mapping measures $\|\nabla f(x_k) + s_{k+1}\|$, whereas RGN measures $\|\nabla f(x_{k+1}) + s_{k+1}\|$ with $s_{k+1} \in \partial h(x_{k+1})$ the subgradient used in the proximal operation.

the proof consists in summing the following interpolation inequalities after multiplication with their respective coefficients. The main difference lies in the choice of the inequalities to be combined; in this proof, we use conditions between consecutive iterates, instead of using conditions between the current iterates and the optimum:

$$\begin{aligned}
f_k \geq f_{k+1} \quad & +\langle g_{k+1}, x_k - x_{k+1}\rangle + \tfrac{1}{2L}\|g_k - g_{k+1}\|^2 \\
& +\tfrac{\mu}{2(1-\mu/L)}\left\|x_k - x_{k+1} - \tfrac{1}{L}(g_k - g_{k+1})\right\|^2 && : \lambda_0 \\
f_{k+1} \geq f_k \quad & +\langle g_k, x_{k+1} - x_k\rangle + \tfrac{1}{2L}\|g_k - g_{k+1}\|^2 \\
& +\tfrac{\mu}{2(1-\mu/L)}\left\|x_k - x_{k+1} - \tfrac{1}{L}(g_k - g_{k+1})\right\|^2 && : \lambda_1, \\
h_k \geq h_{k+1} & + \langle s_{k+1}, x_k - x_{k+1}\rangle && : \lambda_2, \\
h_{k+1} \geq h_k & + \langle s_k, x_{k+1} - x_k\rangle && : \lambda_3.
\end{aligned}$$

We use the following multipliers:

$$\lambda_0 = \lambda_1 = \frac{2}{\gamma}\rho(\gamma) \geq 0, \quad \lambda_2 = \lambda_3 = \frac{2}{\gamma}\rho^2(\gamma) \geq 0.$$

After appropriate substitutions of x_{k+1} and s_*, using $x_{k+1} = x_k - \gamma(g_k + s_{k+1})$ (Section 7.2.2, Condition (a)) and $s_* = -g_*$ (Section 7.2.2, Condition (b)) , we note that the previous weighted sum corresponds to a sum of squares in the two cases of interest (same two regimes as $\rho(\gamma)$, see (RHO)).

◇ When $0 \leq \gamma \leq \frac{2}{L+\mu}$ (i.e., when $\rho(\gamma) = (1 - \gamma\mu)$):

$$\begin{aligned}
(1-\gamma\mu)^2\|g_k + s_k\|^2 \geq & \|g_{k+1} + s_{k+1}\|^2 + (1-\gamma\mu)^2\|s_k - s_{k+1}\|^2 \\
& + \frac{2 - \gamma(L+\mu)}{\gamma(L-\mu)}\|g_k - g_{k+1} - \mu\gamma(g_k + s_{k+1})\|^2, \\
\geq & \|g_{k+1} + s_{k+1}\|^2,
\end{aligned}$$

where the last inequality follows from

$$(1-\gamma\mu)^2 \geq 0, \ 2 - \gamma(L+\mu) \geq 0, \text{ and } \gamma(L-\mu) \geq 0.$$

◇ When $\frac{2}{L+\mu} \leq \gamma \leq \frac{2}{L}$ (i.e., when $\rho(\gamma) = (L\gamma - 1)$):

$$\begin{aligned}
(1-\gamma L)^2\|g_k + s_k\|^2 \geq & \|g_{k+1} + s_{k+1}\|^2 + (1-\gamma L)^2\|s_k - s_{k+1}\|^2 \\
& + \frac{\gamma(L+\mu) - 2}{\gamma(L-\mu)}\|g_k - g_{k+1} - L\gamma(g_k + s_{k+1})\|^2, \\
\geq & \|g_{k+1} + s_{k+1}\|^2,
\end{aligned}$$

and the last inequality follows from

$$(1-\gamma L)^2 \geq 0,\ \gamma(L+\mu)-2 \geq 0,\ \text{and } \gamma(L-\mu) \geq 0.$$

We conclude the proof in the same way as for the distance to optimality: since for any value of γ such that $0 \leq \gamma \leq \frac{2}{L}$, there is always one of the two previous combinations of inequalities that is valid (both multipliers and coefficients of the squared norms are positive), and since the valid one corresponds to the maximum value between the two possible rates $(1-\gamma\mu)^2$ and $(1-\gamma L)^2$, the desired statement is proved.

□

Corollary 7.6. Let $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, and $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ and consider the composite convex optimization problem (7.1) and a feasible starting point $x_0 \in \mathbb{R}^d$ (i.e., x_0 is such that $F(x_0) < \infty$). The iterates of PGM with $0 \leq \gamma \leq \frac{2}{L}$ satisfy:

$$\max_{\substack{f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d) \\ h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d) \\ x_0 \in \mathbb{R}^d, s_0 \in \partial h(x_0)}} \left\{ \frac{\|\nabla f(x_k) + s_k\|^2}{\|\nabla f(x_0) + s_0\|^2} \right\} = \rho^{2k}(\gamma).$$

Proof. Combine Theorem 7.5 with the quadratic lower bounds (QLB). □

Interestingly, the inequalities used in this proof do not involve the optimal point, and only use the information available at the consecutive iterates. In addition, note that as for the convergence in terms of distance to optimality, $\lambda_0 = \lambda_1$ tells us that the result hold under the following weaker assumption:

$$\begin{aligned}\langle g_{k+1} - g_k, x_{k+1} - x_k \rangle \geq & \frac{1}{L}\|g_k - g_{k+1}\|^2 \\ & + \frac{\mu}{1-\mu/L}\left\| x_k - x_{k+1} - \frac{1}{L}(g_k - g_{k+1}) \right\|^2.\end{aligned}$$

A consequence of this inequality is that one can benefit from using the locally better strong convexity and smoothness parameters (i.e., better constants μ and L that satisfy this inequality for two consecutive iterates) instead of the global ones, in order to improve the convergence rate. Also, it is possible to exploit this in order to make online estimations of the strong convexity and smoothness parameters μ and L (we leave this for further research).

Objective function accuracy. Finally, we consider convergence in terms of objective function accuracy. The proof of this convergence rate is much more tedious than the previous ones, and seems to require more assumptions (i.e.,

more inequalities appear to be needed — of course it may be that we just did not isolate the simplest proof).

Theorem 7.7 (Objective function accuracy)**.** Let $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, and $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ and consider the composite convex optimization problem (7.1) and a feasible starting point $x_0 \in \mathbb{R}^d$ (i.e., x_0 is such that $F(x_0) < \infty$). The iterates of PGM with $0 \leq \gamma \leq \frac{2}{L}$ satisfy the following:

$$F(x_{k+1}) - F_* \leq \max\left\{(1-L\gamma)^2, (1-\mu\gamma)^2\right\}(F(x_k) - F_*)\,.$$

Proof. We combine the following interpolation after multiplication with their respective coefficients:

$$\begin{aligned}
f_k \geq f_{k+1} \quad & +\langle g_{k+1}, x_k - x_{k+1}\rangle + \tfrac{1}{2L}\|g_k - g_{k+1}\|^2 \\
& +\tfrac{\mu}{2(1-\mu/L)}\left\|x_k - x_{k+1} - \tfrac{1}{L}(g_k - g_{k+1})\right\|^2 && : \lambda_0,\\
f_* \geq f_k \quad & +\langle g_k, x_* - x_k\rangle + \tfrac{1}{2L}\|g_k - g_*\|^2 \\
& +\tfrac{\mu}{2(1-\mu/L)}\left\|x_k - x_* - \tfrac{1}{L}(g_k - g_*)\right\|^2 && : \lambda_1,\\
f_* \geq f_{k+1} \quad & +\langle g_{k+1}, x_* - x_{k+1}\rangle + \tfrac{1}{2L}\|g_* - g_{k+1}\|^2 \\
& +\tfrac{\mu}{2(1-\mu/L)}\left\|x_* - x_{k+1} - \tfrac{1}{L}(g_* - g_{k+1})\right\|^2 && : \lambda_2,\\
h_k \geq h_{k+1} + \langle s_{k+1}, x_k - x_{k+1}\rangle & && : \lambda_3,\\
h_* \geq h_{k+1} + \langle s_{k+1}, x_* - x_{k+1}\rangle & && : \lambda_4.
\end{aligned}$$

We use the following multipliers:

$$\lambda_0 = \rho(\gamma),\ \lambda_1 = (1-\rho(\gamma))\rho(\gamma),\ \lambda_2 = 1-\rho(\gamma),\ \lambda_3 = \rho^2(\gamma),\ \lambda_4 = 1-\rho^2(\gamma).$$

Appropriate substitutions of x_{k+1} and s_* using $x_{k+1} = x_k - \gamma(g_k + s_{k+1})$ (Section 7.2.2, Condition (a)) and $s_* = -g_*$ (Section 7.2.2, Condition (b)) , we obtain that the weighted sum of inequalities is equivalent to the following expressions.

- ◇ When $0 \leq \gamma \leq \frac{2}{L+\mu}$ (i.e., when $\rho(\gamma) = (1-\gamma\mu)$):

$$\begin{aligned}
&(1-\gamma\mu)^2(F(x_k) - F_*)\\
&\geq F(x_{k+1}) - F_* + \frac{(2-\gamma\mu)\beta}{2\alpha}\|(1-\gamma\mu)g_k - g_{k+1} + \mu\gamma g_*\|^2\\
&+ \frac{\gamma L\mu^2(2-\gamma\mu)}{2(L-\mu)}\left\|(x_k - x_*) - \frac{2L - 2\mu + \gamma\mu^2}{L\mu(2-\gamma\mu)}s_{k+1} - \frac{g_k + g_{k+1}}{\mu(2-\gamma\mu)} + \frac{g_*}{L}\right\|^2\\
&+ \frac{\alpha\gamma\mu}{2L(L-\mu)(2-\gamma\mu)}\left\|s_1 + \frac{(\mu\gamma-1)L\beta}{\alpha}g_k + \frac{L\beta}{\alpha}g_{k+1} + \frac{(L-\mu)(2-\gamma\mu)^2}{\alpha}g_*\right\|^2,\\
&\geq F(x_{k+1}) - F_*, \text{ (because of the signs of the coefficients, see discussion below),}
\end{aligned}$$

with $\alpha = -(\gamma^2 L^2 \mu + 2L(-2+\gamma\mu) + \mu(-2+\gamma\mu)^2)$ and $\beta = (2-\gamma(L+\mu))$. Note that α is positive for $0 \leq \mu < L$ and $0 \leq \gamma \leq \frac{2}{\mu+L}$. Indeed, by denoting

$$-\alpha = p(\gamma) = \gamma^2 L^2 \mu + 2L(-2+\gamma\mu) + \mu(-2+\gamma\mu)^2$$

(positive definite quadratic function), we have $p(0) \leq 0$ and $p(\gamma_c) = -\frac{4L^2(L-\mu)}{(L+\mu)^2} \leq 0$.

- ◇ When $\frac{2}{L+\mu} \leq \gamma \leq \frac{2}{L}$ (i.e., when $\rho(\gamma) = (L\gamma - 1)$):

$$\begin{aligned}
&(1-\gamma L)^2 \left(F(x_k) - F_*\right) \\
&\geq F(x_{k+1}) - F_* + \frac{(2-\gamma L)\beta}{2\gamma\alpha} \|(1-\gamma L)g_k - g_{k+1} + \gamma L g_*\|^2 \\
&+ \frac{\gamma L^2 \mu (2-\gamma L)}{2(L-\mu)} \left\| (x_k - x_*) - \frac{s_{k+1}}{\mu} + \frac{1-\gamma L - \gamma\mu}{\gamma L \mu} g_k - \frac{g_{k+1}}{\gamma L \mu} + \frac{g_*}{L} \right\|^2 \\
&+ \frac{\gamma\alpha}{2\mu(L-\mu)} \left\| s_{k+1} + \frac{(\gamma L - 1)L\beta}{\gamma\alpha} g_k + \frac{L\beta}{\gamma\alpha} g_{k+1} + \frac{(2-\gamma L)(L-\mu)\mu}{\alpha} g_* \right\|^2, \\
&\geq F(x_{k+1}) - F_*, \text{ (because of the signs of the coefficients, see discussion below),}
\end{aligned}$$

with $\alpha = (-2L^2 - 2\mu^2 + 2L\mu + \gamma L^3 + \gamma L \mu^2)$ and $\beta = (\gamma(L+\mu) - 2)$. Again, α is nonnegative as $\alpha = p(\gamma)$ is an increasing linear function which is nonnegative in the region of interest $\gamma \geq \frac{2}{L+\mu}$. Indeed, on can check that by evaluating $p(.)$ at $\gamma_c = \frac{2}{L+\mu}$:

$$p(\gamma_c) = 2\mu^2 \left(\frac{L-\mu}{L+\mu}\right) \geq 0.$$

We conclude the proof in the same way as before: among the two cases, the valid one corresponds to the maximum value between the two possible rates $(1-\gamma\mu)^2$ and $(1-\gamma L)^2$. □

Corollary 7.8. Let $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, and $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ and consider the composite convex optimization problem (7.1) and a feasible starting point $x_0 \in \mathbb{R}^d$ (i.e., x_0 is such that $F(x_0) < \infty$). The iterates of PGM with $0 \leq \gamma \leq \frac{2}{L}$ satisfy the following:

$$\max_{\substack{f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d) \\ h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d) \\ x_0 \in \mathbb{R}^d}} \left\{ \frac{F(x_k) - F(x_*)}{F(x_0) - F(x_*)} \right\} = \rho^{2k}(\gamma).$$

Proof. Combine Theorem 7.7 with the quadratic lower bounds (QLB). □

Proof of Theorem 7.2. The theorem is the combination of Corollary 7.4, Corollary 7.6 and Corollary 7.8.

7.3 Mixed performance measures

In this section, we summarize the global convergence results we obtained on PGM. In order to do that, we first show that Theorem 7.2 can be used to obtain tight bounds on *mixed* performance measures.

Proposition 7.9. Consider the composite convex optimization problem (7.1) and a feasible point $x \in \mathbb{R}^d$ (i.e. x is such $F(x) < \infty$) such that $s \in \partial h(x)$. The following inequalities are satisfied:

(i) $\|x - x_*\|^2 \leq \frac{1}{\mu^2}\|\nabla f(x) + s\|^2$,

(ii) $F(x) - F(x_*) \leq \frac{1}{2\mu}\|\nabla f(x) + s\|^2$.

(iii) $\|x - x_*\|^2 \leq \frac{2}{\mu}(F(x) - F(x_*))$,

Proof. (i) By strong convexity of F, we have

$$\|\nabla f(x) + s - \nabla f(x_*) - s_*\|^2 \geq \mu^2\|x - x_*\|^2,$$

with $s_* \in \partial h(x_*)$ such that $\nabla f(x_*) + s_* = 0$. Therefore

$$\|x - x_*\|^2 \leq \frac{1}{\mu^2}\|\nabla f(x) + s\|^2.$$

(ii) By strong convexity of F (and feasibility of x), we have

$$F(x) - F_* \leq \frac{1}{2\mu}\|\nabla f(x) + s - \nabla f(x_*) - s_*\|^2 = \frac{1}{2\mu}\|\nabla f(x) + s_k\|^2.$$

(iii) Again, by strong convexity of F, we have:

$$F(x) \geq F(x_*) + \langle \nabla f(x_*) + s_*, x - x_* \rangle + \frac{\mu}{2}\|x - x_*\|^2$$

with $s_* \in \partial h(x_*)$ such that $\nabla f(x_*) + s_* = 0$, we obtain the statement. □

Theorem 7.10. Consider the composite convex optimization problem (7.1) and a feasible starting point $x_0 \in \mathbb{R}^d$ (i.e. x_0 is such $F(x_0) < \infty$) such that $s_0 \in \partial h(x_0)$. The iterates of PGM satisfy the following inequalities:

(i) $\|x_k - x_*\|^2 \leq \frac{\rho^{2k}(\gamma)}{\mu^2}\|\nabla f(x_0) + s_0\|^2$,

(ii) $F(x_k) - F(x_*) \leq \frac{\rho^{2k}(\gamma)}{2\mu}\|\nabla f(x_0) + s_0\|^2$,

(iii) $\|x_k - x_*\|^2 \leq \frac{2\rho^{2k}(\gamma)}{\mu}(F(x_0) - F(x_*))$.

Proof. Combine results of Theorem 7.2 with those of Proposition 7.9. □

Note that the global bounds provided by the previous theorem are exact for the step sizes $0 \leq \gamma \leq \frac{2}{L+\mu}$ (thus also $\gamma = \frac{1}{L}$), as provided by the quadratic function $f_\mu(x)$ from Section 7.2.1 (i.e., PGM applied on $F = f_\mu$ satisfies (i),(ii) and (iii) with equalities). Note that the guarantees of Theorem 7.10 do not achieve exactness for larger step sizes. The exact global convergence guarantees are summarized in Table 7.1.

Initialization	$\|x_0 - x_*\|^2$	$F(x_0) - F_*$	$\|\nabla f(x_0) + s_0\|^2$
$\|x_k - x_*\|^2 \leq$	$\rho^{2k}\|x_0 - x_*\|^2$	$\frac{2}{\mu}\rho^{2k}(F(x_0) - F_*)$	$\frac{1}{\mu^2}\rho^{2k}\|\nabla f(x_0) + s_0\|^2$
$F(x_k) - F_* \leq$	$\star$	$\rho^{2k}(F(x_0) - F_*)$	$\frac{1}{2\mu}\rho^{2k}\|\nabla f(x_0) + s_0\|^2$
$\|\|\tilde{\nabla} f(x_k)\|\|^2 \leq$	$\star$	$\star$	$\rho^{2k}\|\nabla f(x_0) + s_0\|^2$

Table 7.1: Summary of the global convergence guarantees proposed by Theorem 7.2 (exact) and Theorem 7.10 (exact for $0 \leq \gamma \leq \frac{2}{L+\mu}$). The corresponding results for the case of f being quadratic and $h = 0$ are presented in the work of Nemirovski [Nem92]. The stars denote the combinations of performance measures for which no analytical *global and exact* convergence guarantees were obtained yet.

7.4 Conclusion

Tight convergence rates for PGM. The main contribution of this work is to close the gap between lower and upper complexity bounds for PGM in smooth strongly convex optimization. We obtained exact global linear convergence rates for measuring progress in terms of different measures of optimality.

The proof methodology used in order to prove the main results allows a clear and transparent use of the assumptions of the theorems. As an example, we observed that strong convexity was only required between certain pairs of points.

In addition, Theorem 7.2 may be used to extend the recent results of [dKGT16, Theorem 1.2] on the exact worst-case complexity of the gradient descent with exact line search. Furthermore, as in the unconstrained case ($h(x) = 0$), this result cannot be improved in general, as it is attained by a two-dimensional quadratic example [dKGT16, Example 1.3].

Proximal gradient method with exact line search

Input: $x_0 \in \mathbb{R}^d$, $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$, $0 \leq \gamma \leq \frac{2}{L}$.

For $k = 0 : N-1$

$$\gamma = \operatorname*{argmin}_{\gamma \in \mathbb{R}} F\left[p_{\gamma h}\left(x_k - \gamma \nabla f(x_k)\right)\right]$$

$$x_{k+1} = p_{\gamma h}\left(x_k - \gamma \nabla f(x_k)\right)$$

Corollary 7.11. Let $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, and $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ and consider the composite convex optimization problem (7.1) and a feasible starting point $x_0 \in \mathbb{R}^d$ (i.e., x_0 is such that $F(x_0) < \infty$). The iterates of PGM with exact line search satisfy the following inequality:

$$F(x_{k+1}) - F_* \leq \left(\frac{L-\mu}{L-\mu}\right)^2 \left(F(x_k) - F_*\right).$$

Proof. This is exactly the result of Theorem 7.7 with $\gamma = \frac{2}{L+\mu}$. The corresponding result is an upper bound on the worst-case of PGM with exact line search, which turns out to be tight on the Example 6.2 (quadratic example in [Ber99, Example on p.69] or [dKGT16, Example 1.3]). □

Further work. Tight convergence results are still open for a variety of first-order methods and different convergence measures, as for example for accelerated schemes [Nes04], for inexact methods [DGN14, SLRB11] and coordinate descent schemes [Nes12a]. Obtaining such tight convergence results opens the door for a better use of gradient-schemes as primitive operations in more complicated algorithms, but also for designing optimized first-order methods (such research directions are carried out among others in [DT14, KF16d] for the smooth unconstrained convex case and in [LRP16] in the strongly convex case with the presence of disturbing noise).

Part III

Conclusion

Chapter 8

Further Developments in Performance Estimation

In this chapter, we broaden the applicability range of performance estimation. We do not intend to give a complete overview of the literature for the corresponding applications, but rather provide a variety of ideas for further developments. In addition we use those examples to emphasize the difficulties and limitations inherent to performance estimation and which require further investigation. Those can be summarized in three points

- ◇ first, finding interpolation conditions the problems of interest is most often far from being trivial. We provide examples for which no tractable interpolation conditions were found. Solving the corresponding (relaxed) performance estimation problems therefore results in upper bounds.
- ◇ Second, finding tractable formulations of the algorithms compatible with the representation of the class of functions may be involved, and also generally results in relaxations and upper bounds.
- ◇ Finally, the computational cost for solving the corresponding semidefinite programs may become prohibitive, even in apparently very simple situations.

Note that we are only interested in solving the performance estimation problems to global optimality (or, at least, to obtain guaranteed upper bounds), hence our interest in convex formulations. However, one should note that any nonconvex formulation that can be solved to global optimality (or for which meaningful upper bounds can be obtained) can also be used.

The chapter is organized as follows:

- ◇ in Section 8.1, we illustrate that standard algorithms for solving monotone inclusions very naturally fit into the performance estimation framework.

- ◇ Then, we provide other examples of applications of the performance estimation framework for decentralized, randomized, nonconvex and noisy optimization problems in Section 8.2. In addition, we introduce several open problems, including the treatment of second-order methods and of non-Euclidean geometries.
- ◇ In Section 8.3, we finally conclude the chapter by emphasizing the main advantages, difficulties and aspects that require further attention for a broader applicability of the performance estimation framework in various practical settings.

8.1 Monotone operators and splitting methods

The goal of this section is to convince the reader that the performance estimation framework naturally applies to *monotone operators*. We illustrate our point on the forward-backward and Douglas-Rachford splitting schemes, after reviewing some of the basic underlying concepts of the field. For a very nice introduction to monotone operators, we refer the reader to the thesis of Eckstein [Eck89] and to the seminal references [RW98, BC11].

Monotone operators are a natural extension to subdifferentials (other examples can be found e.g., in the tutorial [RB16]). They are more and more used in the mathematical optimization community, in part because of their natural ability for developing distributed algorithms. As an example, the very popular alternating direction method of multipliers (ADMM) [BPC$^+$11] can be seen as a dual version of the Douglas-Rachford splitting scheme (see e.g., [Gab83, Eck89]).

Notations 8.1. We denote by $2^{\mathbb{R}^d}$ the set of all subsets of $\mathbb{R}^d$. We use the notation $T : \mathbb{R}^d \to 2^{\mathbb{R}^d}$ for meaning that T is a set-valued operator from $\mathbb{R}^d$ to $\mathbb{R}^d$. That is, the operator T maps every $x \in \mathbb{R}^d$ to a set $Tx \subseteq \mathbb{R}^d$. A convenient way to characterize T is via its graph:

$$\operatorname{graph} T = \left\{(x, z) \in \mathbb{R}^d \times \mathbb{R}^d \mid z \in Tx\right\}.$$

Definition 8.2. An operator $T : \mathbb{R}^d \to 2^{\mathbb{R}^d}$ is monotone if $\forall (x_1, z_1), (x_2, z_2) \in \operatorname{graph} T$ we have

$$\langle z_1 - z_2, x_1 - x_2 \rangle \geq 0.$$

Definition 8.3. A monotone operator $T : \mathbb{R}^d \to 2^{\mathbb{R}^d}$ is maximally monotone if there is no monotone operator S such that graph $T \subsetneq$ graph S.

Note that it is well known that requiring an operator T to be monotone is only a necessary condition for the existence of a convex function f satisfying $\partial f = T$. In order to obtain a sufficient condition, one needs to consider cylic

monotonicity conditions[1].

Zeros of maximally monotone operators. The general problem of interest in this section is to solve the following monotone inclusion:

$$\text{find } x \in \mathbb{R}^d \text{ such that } 0 \in T(x),$$

with $T : \mathbb{R}^d \to 2^{\mathbb{R}^d}$ some maximally monotone set-valued operator. The common approach in the monotone operator framework is to reformulate this as an equivalent fixed-point problem. One way to proceed is to associate T with its *resolvent* $J_{\lambda T} : \mathbb{R}^d \to \mathbb{R}^d$

$$J_{\lambda T} = (I + \lambda T)^{-1},$$

where I is the identity operator. It can be shown (see e.g., [Eck89, Proposition 3.9]) that for any maximally monotone operator T, $\forall \lambda > 0$ and $\forall x \in \mathbb{R}^d$:

$$J_{\lambda T}(x) = \{x\} \Leftrightarrow 0 \in T(x).$$

The resolvent $J_{\lambda T}$ has numerous nice properties due to the maximal monotonicity of T; among others, it is *(firmly) nonexpansive* (this motivates using fixed-point iterations, see e.g., [Eck89, Definition 3.13]) and satisfies $\operatorname{dom} J_{\lambda T} = \mathbb{R}^d$ (that is, $\forall x \in \mathbb{R}^d$ we have $J_{\lambda T}(x) \neq \emptyset$, which is very important for practical computations). Also, note that the fixed-point iteration

$$x_{k+1} = J_{\lambda T}(x_k) \Leftrightarrow x_{k+1} + \lambda T(x_{k+1}) = x_k$$

is usually referred to as the proximal point algorithm; which we already studied in Section 5.3.1 when $T = \partial f$ for some $f \in \mathcal{F}_{0,\infty}$.

Interpolation of maximally monotone operators. In order to use the performance estimation framework with tightness guarantees, we consider the interpolation problem for maximally monotone operators. That is, given an index set and set of couples $S = \{(x_i, t_i)\}_{i \in I}$ with $x_i, t_i \in \mathbb{R}^d$ $\forall i \in I$, we want to find conditions guaranteeing the existence of a maximally monotone operator T such that

$$t_i \in T(x_i) \ \forall i \in I.$$

For that purpose, we define the operator $T_S : \mathbb{R}^d \to 2^{\mathbb{R}^d}$

$$T_S(x) = \{g \in \mathbb{E}^* \mid (x, g) \in S\}.$$

[1] See the related convex integration problem from Section 3.4, and the references [RW98, Theorem 12.15] and [BC11, Theorem 22.14].

Note that if the set S satisfies discrete monotonicity conditions, i.e., if

$$\langle t_i - t_j, x_i - x_j \rangle \geq 0 \ \forall i, j \in I, \tag{8.1}$$

then the operator T_S is monotone. In addition, the following theorem implies the existence of a maximally monotone extension of T_S.

Theorem 8.4. [BC11, Theorem 20.21] Let $T : \mathbb{R}^d \to 2^{\mathbb{R}^d}$ be monotone. Then there exists a maximally monotone extensions of T; i.e., a maximally monotone operator $\tilde{T} : \mathbb{R}^d \to 2^{\mathbb{R}^d}$ such that $\operatorname{graph} T \subseteq \operatorname{graph} \tilde{T}$.

Note that this theorem is proved using Zorn's lemma (equivalent to the axiom of choice), which is actually a key element in monotone operator theory (see e.g., discussion in [Eck89, Section 2.1]). Therefore, tightness of the results for monotone operators relies on the axiom of choice — however, we suspect there is a constructive way not relying on Zorn's lemma as the set S is finite.

Before going into the next section, let us mention the existence of strong monotonicity and cocoercivity conditions. Those conditions are respectively corresponding to strong convexity and smoothness in the case of proper closed convex functions. The definitions are the following: $\forall x_1, x_2 \in \mathbb{R}^d$, $t_1 \in Tx_1$, $t_2 \in Tx_2$ we have

$$\begin{aligned} &\langle t_1 - t_2, x_1 - x_2 \rangle \geq \frac{1}{L}\|t_1 - t_2\|^2, && (1/L\text{-cocoercivity}),\\ &\langle t_1 - t_2, x_1 - x_2 \rangle \geq \mu\|x_1 - x_2\|^2, && (\mu\text{-strong monotonicity}). \end{aligned}$$

As in the case of convex functions, one can easily develop interpolation conditions for dealing with $1/L$-cocoercivity and μ-strong monotonicity. For doing that, one possibility is exploit the following elements:

- ◇ A maximally monotone operator $T(x)$ is μ-strongly monotone if and only if $(T - \mu I)(x)$ is maximally monotone.
- ◇ A maximally monotone operator $T(x)$ is $1/L$-cocoercive if and only if $T^{-1}(x)$ is L-strongly monotone and maximal.

Note that this does not provide a way to interpolate an operator being both strongly monotone and cocoercive, which we leave as an open question.

Splitting methods. For the following examples, let us consider the following monotone inclusion

$$\text{find } x \in \mathbb{R}^d \text{ such that } 0 \in A(x) + B(x), \tag{MI}$$

with $A, B : \mathbb{R}^d \to 2^{\mathbb{R}^d}$ being a maximally monotone operators. In addition, we assume $A = \partial f$ for some $f \in \mathcal{F}_{\mu,L}$ and we use the interpolation conditions from Theorem 3.8 for the set $\{(x_i, a_i, f_i)\}$ in the following examples.

Forward-backward splitting. The forward-backward splitting (FBS) scheme for solving (MI) performs the following fixed-point iterations:

$$x_{k+1} = J_{\lambda B}[(I - \lambda A)x_k].$$

One can note that the performance estimation problems corresponding to this scheme have the same forms as those obtained from the proximal gradient methods from Chapter 7 as long as no function values are involved (as B is a general monotone operator). Therefore, the results in terms of gradient norms and distances to the solution (Theorem 7.3 and Theorem 7.5) can be adapted to FBS:

$$\|x_{k+1} - x_*\|^2 \leq \max\{(1-\lambda\mu)^2, (1-\lambda L)^2\}\|x_k - x_*\|^2,$$
$$\|A(x_{k+1}) + b_{k+1}\|^2 \leq \max\{(1-\lambda\mu)^2, (1-\lambda L)^2\}\|A(x_k) + b_k\|^2,$$

with x_* the (unique) solution to the monotone inclusion $0 \in A(x_*) + B(x_*)$, some $b_k \in B(x_k)$ and $b_{k+1} \in B(x_{k+1})$ the (unique) vector used in the iteration

$$x_{k+1} + \lambda b_{k+1} = x_k - \lambda A(x_k).$$

Douglas-Rachford splitting. For solving (MI), the Douglas-Rachford splitting[2] (DRS) uses the alternative fixed-point iterations:

$$w_{k+1} \in J_{\lambda A}[(2J_{\lambda B} - I)w_k] + [I - J_{\lambda B}](w_k),$$

which converges to some w_* such that $w_* = (2J_{\lambda A} - I)(2J_{\lambda B} - I)w_*$. In order to obtain the corresponding x_* such that $0 \in A(x_*) + B(x_*)$, we use the equivalence:

$$0 \in A(x_*) + B(x_*) \Leftrightarrow x_* = J_{\lambda B} w_*.$$

Note that DRS can be written in an expanded form; find x_{k+1}, $B(x_{k+1})$, y_{k+1} and $A(y_{k+1})$ such that:

$$x_{k+1} = w_k - \lambda B(x_{k+1}),$$
$$y_{k+1} = 2x_{k+1} - w_k - \lambda A(y_{k+1}),$$
$$w_{k+1} = y_{k+1} - x_{k+1} + w_k.$$

With little effort, this method can be written using the (FSLFOM) format from Chapter 5. As an example, the framework easily allows obtaining the following result (proofs similar to that of Chapter 6 and Chapter 7):

$\|w_{k+1} - w_*\| \leq \max\left\{\frac{1}{1+\lambda\mu}, \frac{\lambda L}{\lambda L+1}\right\}\|w_k - w_*\|$ (Note that this observation reproduces the recent results of [GB16, Theorem 2]).

[2] See [LM79, Algorithm II] for the original presentation.

We leave the study of other settings with and without linear convergence for further work (as for example tightening the results of [Gis15, Theorem 1] for when A is strongly monotone and B is cocoercive).

8.2 Further developments

In this section, we further illustrate potential uses of the performance estimation framework. We do not provide technical details but rather focus on convincing the reader of the broader applicability of the framework.

Decentralized gradient methods. Consider the following composite optimization problem

$$\min_{x_i \in \mathbb{R}^d} \left\{ \sum_{i=1}^{n} f_i(x_i) \quad \text{s.t. } x_i = x_j \ \forall i, j = 1, \dots, n \right\},$$

with $f_i \in \mathcal{F}_{\mu_i, L_i}(\mathbb{R}^d)$. When the computations have to be performed among n computers, each one storing information about a single function f_i, one possibility is to use a consensus-based (sub)gradient method (see e.g., [JKJJ08, NOP10])

$$\begin{pmatrix} x_1^{k+1} \\ \vdots \\ x_n^{k+1} \end{pmatrix} = W \begin{pmatrix} x_1^k - \gamma_1^k \tilde{\nabla} f_1(x_1^k) \\ \vdots \\ x_n^k - \gamma_n^k \tilde{\nabla} f_n(x_n^k) \end{pmatrix},$$

with W being some known doubly stochastic matrix. This model is a particular instance of (FSLFOM), and can therefore be modelled in the performance estimation framework with tightness guarantees. Note however that the size of the problems grows linearly in both the number of components n and in the number of iterations N.

Randomized methods. Let us consider two kinds of randomized methods: stochastic gradient and block-coordinate descent. In those cases, the performance estimation approach can be used to search for functions with the worst *expected* convergence results. Indeed, one can apply the PEP framework to

- ◇ the study of stochastic gradient descent on the n-term objective function

$$\min_{x \in \mathbb{R}^d} \sum_{i=1}^{n} f_i(x),$$

with $f_i \in \mathcal{F}_{\mu_i, L_i}$ (see e.g., [Ber10, SSBD14, Kiw04]), with the corresponding update $x_{k+1} = x_k - \gamma_{k,i} \nabla f_i(x_k)$ with probability p_i.

◇ The application of randomized block-coordinate descent on the problem

$$\min_{x^{(i)} \in \mathbb{R}^{d_i}} f(x^{(1)}, \ldots, x^{(n)}),$$

with $f \in \mathcal{F}_{0,\infty}$ and being L_i-smooth separately on every block of coordinates $x^{(i)}$ (see e.g., [RT14, FR15, Nes12a]), and the corresponding iteration with probability p_i:

$$\begin{cases} x_{k+1}^{(j)} = x_k^{(j)} & \text{for } i \neq j, \\ x_{k+1}^{(i)} = x_k^{(i)} - \gamma_k^{(i)} \nabla_i f(x_k), & \text{otherwise,} \end{cases}$$

(the standard notation $\nabla_i f(x)$ is used for meaning $\frac{\partial f}{\partial x^{(i)}}(x)$).

In both cases, the algorithm perform updates on one of the components (randomly selected) at each iteration — see Figure 8.1 for $n = 2$. In order to evaluate the expectation of some convergence measure after N iterations, we (a priori) have to average over all possible sequences of choices, which results in an exponential number of combinations: $(n^{N+1} - 1)/(n - 1)$; this renders the corresponding SDP intractable even for relatively small values of n and N.

Also note that in the case of block coordinate descent methods, it is usual to assume that the convex function to be minimized satisfies a smoothness condition separately on every block of coordinates. It is not clear how to interpolate on this class of convex and *block-smooth* functions. For example, the discrete version of the conditions

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \frac{1}{2L_i} \|\nabla_i f(x) - \nabla_i f(y)\|^2$$

are used in [SL16, Lemma 1.1] for studying cyclic block coordinate descent schemes; however, it appears that they do not correspond to interpolation conditions.

Nonconvexity. As outlined in Chapter 3, it is also possible to use the performance estimation framework for analyzing first-order algorithms for nonconvex functions. For example, one can use the nonconvex smooth interpolation result from Theorem 3.21 for studying algorithms tailored for

$$\min_{x \in \mathbb{R}^d} f(x),$$

when f is a L-smooth nonconvex function. However, this is generally difficult to combine with the expression of the algorithm in consideration. We illustrate the potential difficulties on the following example.

First, for the steepest descent with exact line-search, we only managed to re-

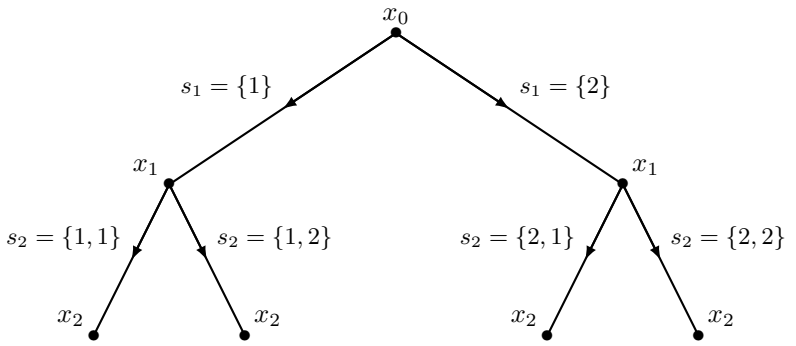


Figure 8.1: Possible sequences obtained from a 2-terms objective function (for stochastic gradient descent) or from a 2-blocks coordinate descent scheme. The list s_i denotes the ordered list of updated components done so far.

produce the following well known result (see e.g., [Nem99, Proposition 3.3.1]):

$$\min_{0\leq i\leq N} \|\nabla f(x_i)\|_{\mathbb{E}^*}^2 \leq \frac{2}{N}(f(x_0) - f(x_N)) \left(\leq \frac{2}{N}(f(x_0) - f(x_*))\right).$$

In that context, we identify two main difficulties:

- the intermediate optimality conditions can not be expressed easily (zero gradient on a line is not sufficient to guarantee optimality),
- expressing the search direction is a nonconvex constraint (in the smooth strongly convex case, however it appeared that using a relaxation was sufficient — see Chapter 6).

Note that this is consistent with the fact exact line search can hardly be used in practice without additional assumptions on the function f, as we can generally only have the guarantee of finding critical points.

Nevertheless, we can still consider fixed-step gradient methods (we use step size $1/L$) with tightness guarantees — even if it is much less practical in the general nonconvex setting. In that situation, it is possible to obtain the following corresponding convergence result for minimizing an L-smooth nonconvex function with no constraint (using smooth nonconvex interpolation conditions as in Theorem 3.21):

$$\min_{0\leq i\leq N} \|\nabla f(x_i)\|_{\mathbb{E}^*}^2 \leq \frac{4}{3N}(f(x_0) - f(x_N)) \left(\leq \frac{4}{3N}(f(x_0) - f(x_*))\right).$$

Inexact computations. In Section 5.2.3, we briefly talked about handling inexact methods using performance estimation. As underlined in that section, not all inexactness models can be handled directly with tightness guarantees. As example, the basic noise models as used in Theorem 5.15, or the model proposed by d'Aspremont [d'A08] can easily be used in the framework. However, inexact (δ, L)-oracles developed by Devolder, Glineur and Nesterov [DGN14] do not appear to have (as such) corresponding interpolation conditions (so only upper bounds are usually found).

Note that all the previous results may easily be studied (at least numerically) in the presence of such noise models (e.g., splitting methods with inexact oracles).

Further ideas. For further development, we believe there are still a lot of different possibilities for exploring numerical optimization schemes in a tight way using the performance estimation approach. Among others, the previous chapters did not mention the following possible further directions.

- ◇ Structure is crucial for obtaining efficient algorithms (see e.g., [Nes08]). In this work, we mainly considered generic functions on which it is easy to perform some operations (oracles); however, considering for example more specifically linear or quadratic functions as parts of the objective function may provide more appropriate results when dealing with practical problems for which we specifically know the structure.
- ◇ As previously underlined, choosing the appropriate problem formulation/structure can play a crucial role in our ability to solve it. In particular, we restricted ourselves to Euclidean norms for the whole work, whereas it may be more appropriate to adapt the choice of the setting (e.g., the norm in which we require smoothness and strong convexity) to the domain. For example, this is the underlying idea behind the Mirror Descent algorithm [BT03, BTN01] (note that the PEP framework easily allows dealing with Bregman divergences, but as far as we know, cannot handle strong convexity and/or smoothness with respect to non-Euclidean norms other than by using their equivalence with Euclidean ones).
- ◇ In addition to the previous point, we suggested the use of PEP for second order methods in Chapter 5. However, we did not study them so far — we believe this could be done using the standard idea of measuring progress in terms of the local norm induced by the Hessian (which is among the motivations for using generic Euclidean norms in most of the work). In addition, we believe that attention should be given to the analysis of quasi-Newton-type methods (see e.g., [NW06]) using the PEP framework.

8.3 Conclusion

Before going into the final concluding chapter, let us summarize what we learned. First, there is still room for using the performance estimation framework to improve, or develop the analyses of a lot of practical optimization schemes, including currently very popular splitting methods.

Second, the performance estimation framework provide a generic methodology for analyzing optimization schemes in a tight way. However, the requirements for obtaining a tight analysis are generally difficult to satisfy (i.e., obtain appropriate interpolation conditions, formulate the algorithm in a tractable way). Nevertheless, meaningful relaxations of PEPs may in general still provide interesting convergence results — we can generally at least match the previous known results.

Finally, desired developments include the possibility of handling second-order and quasi-Newton methods, and to handle non-Euclidean geometries. Those potential developments are apparently of very different natures when compared to the methods studied using the PEP framework so far, and may therefore require a significant amount of work.

Chapter 9

Research Outcomes and Perspectives

Research outcomes

Contributions to worst-case analyses. The main contribution of this work was the development of the performance estimation framework, whose aim is to automate the generation of worst-case guarantees for a family of optimization algorithms. The core underlying idea is to use (convex) optimization to perform worst-case analyses of optimization schemes. The fundamentals are summarized in the following points.

- ◇ This framework uses optimization software to perform worst-case analyses. The idea is to generate problem instances on which the algorithm under consideration behaves as bad as possible. The worst-case computation problem is itself an optimization problem whose variables are an objective function and a domain. Hence, the worst-case computation is an *infinite-dimensional* optimization problem.
- ◇ Under the assumption that the algorithm evaluate the function and its gradient at a finite set of points (black-box/oracle assumption), the worst-case computation (or performance estimation) problem can be formulated in a *finite* way using an appropriate discretization.

  In order to render this formulation tractable (more precisely: convex), we use a standard *Gram-matrix* trick — a very standard tool for approximating solutions to NP-hard problems (see e.g., the seminal [GW95]). Note that the Gram-matrix trick is usually used to perform relaxations, which in our case turned out to be very advantageous: it renders the solutions independent of the dimension of the initial decision space.

- ◇ Primal solutions to the performance estimation problems correspond to lower bounds (i.e., actual instances of optimization problems on which the algorithm behaves as badly as possible).
- ◇ Dual solutions to the performance estimation problems correspond to proofs, which can be converted into linear combinations of valid inequalities (i.e., a certification that the primal bound is optimal).

Advantages of the approach. Let us now quickly summarize the main advantages of the approach:

- ◇ it helps designing analytical proofs (both for lower and upper bounds) and to develop the underlying intuitions.
- ◇ It allows to very easily test new assumptions, by just adding the corresponding constraints to the performance estimation problems.
- ◇ It allows designing new methods.

Also, note that the *convex interpolation* framework has the huge advantage of providing sufficient requirements for guaranteeing the existence of (tight) convergence proofs. That is, given an algorithm, a class of functions, a performance measure and an initial condition that can be expressed in the PEP framework, we know it is possible to derive tight convergence proofs with no other (in)equalities characterizing the functions than the interpolation ones.

Limitations of the approach. However, any user of the performance estimation approach should be aware of the following difficulties and limitations.

- ◇ The size of corresponding semidefinite program grows with the number of iterations to be analyzed. Therefore, performance estimation problems can only be solved for limited numbers of iterations. As an example, in the case of first-order methods for smooth unconstrained minimization, we were not able to perform the numerical worst-case computations for more than around 150 iterations on a simple desktop computer.

  The alternative approach of Lessard, Recht and Packard [LRP16] alleviates this difficulty in the case of linearly-converging algorithms, by using Lyapunov stability theory coupled with relaxations and time-invariant algorithms, at the cost of losing tightness guarantees (see discussion in Section 1.3.2).

- ◇ The performance estimation approach is intrinsically conservative when it comes to comparing with practical performances. It is due to the fact we perform worst-case analysis, and especially to the fact we perform it on an iteration-per-iteration basis, which means that there may be no function achieving the worst-case for all number of iterations (so the

actual convergence rates can only be strictly better). Of course, this is a limitation of most standard approaches to worst-case analyses.

Interestingly, this drawback is also partially alleviated in recent works (see e.g., [AP15]). Roughly speaking, we approached the convergence rate by trying to answer the question *what can we exactly guarantee after N iterations ?* (no matter the function, as long as it is within some predetermined class), whereas an alternative approach may ask the following: *given a function, how does the convergence measure improve from iteration to iteration?* This question is approached asymptotically by Attouch and Peypouquet in [AP15]. In the case of the minimization of a smooth convex function $f \in \mathcal{F}_{0,L}$, the result of Attouch and Peypouquet in [AP15, Theorem 1] can be stated as follows: the iterates of a variant of the fast gradient method satisfy

$$\lim_{N\to\infty} N^2(f(x_N) - f(x_*)) = 0,$$

which shows that the asymptotic rate of convergence of the method is strictly better than the well known $\mathcal{O}(N^{-2})$. As far as we know, this kind of results cannot be obtained by standard worst-case approaches as the lower bound proposed in [Nes04, Theorem 2.1.6] is $\mathcal{O}(N^{-2})$.

Another potential approach for going beyond worst-case analysis is the celebrated smoothed analysis technique [ST04]. The underlying idea is to study the *stability* of the problems on which the algorithm of interest achieves its worst-case. We are not aware of applications of this method for worst-case analyses of first-order methods (and we have no idea whether it is relevant to do it), but automated worst-case analyses may be useful for using those techniques (by carefully choosing appropriate types of perturbations).

◇ The analyses obtained by the performance estimation approach are very sensitive to the initial knowledge/assumptions and to choice of the convergence measure. In practice, we generally know more than what is usually assumed (e.g., on the distance to optimality $\|x_0 - x_*\|$), and we should therefore combine different sorts of initial assumptions in order to better link theoretical convergence results with practical observations (see e.g., Table 7.1).

Although this point may seem quite theoretical, the choice of the setting is critical in the development of optimized methods (see [DT14, KF16a, KF16b, KF16c, KF16d]), which are tailored for very specific set of convergence measure and initial condition.

Convex interpolation. Other than performance estimation, a great deal of this work is devoted to convex interpolation (see Chapter 3). Interpolation turned out to play a crucial role in the development of the performance estimation framework. However, we did not explore further the possibilities for using it in other contexts.

Regression schemes represents a major part of optimization problems arising in practice. Under some circumstances, the approximating function is required to be convex (e.g., in economy, for circuit design; see [HD13] and references therein). Although the problem of interpolating a function under convexity/concavity constraint is not new (see e.g., [Hil54]), methods for doing it efficiently were only recently being developed (see e.g., [HD12, HD13, MB09]). In those recent works, the main idea is to represent convex functions through supporting hyperplanes (whereas older works were considering a function structure and imposing convexity of that structure). This kind of representation corresponds to our discrete representation of convex functions from Chapter 3. As some interest towards smooth convex interpolation was also recently raised, (see [AFM11]), we expect our interpolation schemes to be easily transposable to that field.

Technical open questions and perspectives

In the previous sections, we focused on perspectives for further developing the range of applications of performance estimation problems (more classes of functions and algorithms). Let us now consider more general questions and perspectives concerning the treatment of performance estimation problems which are currently missing in the literature.

- ◇ The structure of the SDP should be exploited: when does there exist low-rank solutions (using for example the geometry of the PSD cone [Pat98, Bar01]) ? How to use the structure of PEP arising in high-dimensional contexts (e.g., for randomized methods) ? What can we say *a priori* on the worst-case solutions ? We refer to Remark 4.6 for a short discussion on related topics.

- ◇ Exploit interpolation conditions (and semidefinite programming) for designing new methods. The main difficulty arising in those perspectives is the nonconvexity of the method optimization problem.

  For example, new geometric versions of accelerated methods for smooth strongly convex unconstrained minimization were developed in [DFR16, BLS15], which we believe could be (at least theoretically) refined using appropriate interpolation conditions (see e.g., Example 3.16 and Example 3.29).

  Note that a performance estimation-like approach was developed in [DT16] for designing a new optimal method for nonsmooth convex minimization.

- ◇ In the very recent work [Dro16], a new lower bound was developed using a performance estimation-related approach. This new bound allowed to prove optimality of the optimized gradient method in a particular setting (best worst-case in function values, starting from an initially bounded distance to optimality). We believe that such a methodology could be applied to much more problem classes.

- ◇ Finally, there remains many open questions concerning the worst-case behavior of widely-used optimization schemes (e.g., for splitting methods). In that direction, further extending the framework to handle both larger classes of functions and larger classes of algorithms should clearly be considered.

  However, there may be no way of formulating a PEP in a tractable way for some classes of practical optimization algorithms. Relaxation strategies for finding approximate optimal solutions (while keeping upper bound properties) to nonconvex problems should therefore also be considered in that perspective.

Bibliography

[AFM11] Néstor Aguilera, Liliana Forzani, and Pedro Morin. On uniform consistent estimators for convex regression. *Journal of Nonparametric Statistics*, 23(4):897–908, 2011.

[AP15] Hedy Attouch and Juan Peypouquet. The rate of convergence of Nesterov's accelerated forward-backward method is actually $o(k^{-2})$. *preprint arXiv:1510.08740*, 2015.

[APR15] Hedy Attouch, Juan Peypouquet, and Patrick Redont. Fast convergence of an inertial gradient-like system with vanishing viscosity. *preprint arXiv:1507.04782*, 2015.

[AZO14] Zeyuan Allen-Zhu and Lorenzo Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. *preprint arXiv:1407.1537*, 2014.

[Bar01] Alexander Barvinok. A remark on the rank of positive semidefinite matrices subject to affine constraints. *Discrete & Computational Geometry*, 25(1):23–31, 2001.

[BB08] Olivier Bousquet and Léon Bottou. The tradeoffs of large scale learning. In *Advances in neural information processing systems*, pages 161–168, 2008.

[BC11] Heinz H Bauschke and Patrick L Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, 2011.

[BD86] James P Boyle and Richard L Dykstra. A method for finding projections onto the intersection of convex sets in hilbert spaces. In *Advances in order restricted statistical inference*, pages 28–47. Springer, 1986.

[Bec07] Amir Beck. Quadratic matrix programming. *SIAM Journal on Optimization*, 17(4):1224–1238, 2007.

[Ber99] Dimitri P Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, 1999.

[Ber09] Dimitri P Bertsekas. *Convex Optimization Theory*. Athena Scientific, 2009.

[Ber10] Dimitri P Bertsekas. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optimization for Machine Learning*, pages 1–38, 2010.

[Ber15] Dimitri P Bertsekas. *Convex Optimization Algorithms*. Athena Scientific, 2015.

[BL10] Jonathan M Borwein and Adrian S Lewis. *Convex analysis and nonlinear optimization: theory and examples*. Springer Science & Business Media, 2010.

[BLS15] Sébastien Bubeck, Yin Tat Lee, and Mohit Singh. A geometric alternative to nesterov's accelerated gradient descent. *preprint arXiv:1506.08187*, 2015.

[BM03] Samuel A Burer and Renato D C Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

[BPC+11] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[BT03] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[BT09a] Amir Beck and Marc Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *Image Processing, IEEE Transactions on*, 18(11):2419–2434, 2009.

[BT09b] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[BT12] Amir Beck and Marc Teboulle. Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization*, 22(2):557–580, 2012.

[BTN01] Ahron Ben-Tal and Arkadi S Nemirovski. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*, volume 2. SIAM, 2001.

[Bub15] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[BXM03] Stephen Boyd, Lin Xiao, and Almir Mutapcic. Subgradient methods. *lecture notes of EE392o, Stanford University, Autumn Quarter*, 2003.

[CP11] Patrick L Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

[Cro77] Jean-Pierre Crouzeix. A relationship between the second derivatives of a convex function and of its conjugate. *Mathematical Programming*, 13(1):364–365, 1977.

[d'A08] Alexandre d'Aspremont. Smooth optimization with approximate gradient. *SIAM Journal on Optimization*, 19(3):1171–1183, 2008.

[Dan98] George B Dantzig. *Linear programming and extensions*. Princeton university press, 1998.

[DFR16] Dmitriy Drusvyatskiy, Maryam Fazel, and Scott Roy. An optimal first order method based on optimal quadratic averaging. *preprint arXiv:1604.06543*, 2016.

[DGN12] Olivier Devolder, François Glineur, and Yurii Nesterov. Double smoothing technique for large-scale linearly constrained convex optimization. *SIAM Journal on Optimization*, 22(2):702–727, 2012.

[DGN14] Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1-2):37–75, 2014.

[DK10] Etienne De Klerk. Exploiting special structure in semidefinite programming: A survey of theory and applications. *European Journal of Operational Research*, 201(1):1–10, 2010.

[dKGT16] Etienne de Klerk, François Glineur, and Adrien B Taylor. On the worst-case complexity of the gradient method with exact line search for smooth strongly convex functions. *preprint arXiv:1606.09365 (Accepted in Optimization Letters)*, 2016.

[Dro14] Yoel Drori. *Contributions to the Complexity Analysis of Optimization Algorithms*. PhD thesis, Tel-Aviv University, 2014.

[Dro16] Yoel Drori. The exact information-based complexity of smooth convex minimization. *Journal of Complexity*, 2016.

[DT14] Yoel Drori and Marc Teboulle. Performance of first-order methods for smooth convex minimization: a novel approach. *Mathematical Programming*, 145(1-2):451–482, 2014.

[DT16] Yoel Drori and Marc Teboulle. An optimal variant of kelley's cutting-plane method. *Mathematical Programming*, 160(1):321–351, 2016.

[Eck89] Jonathan Eckstein. *Splitting methods for monotone operators with applications to parallel optimization*. PhD thesis, Massachusetts Institute of Technology, 1989.

[FB53] Werner Fenchel and Donald W Blackett. *Convex cones, sets, and functions*. Princeton University, Department of Mathematics, 1953.

[Fen49] Werner Fenchel. On conjugate convex functions. *Canadian Journal of Mathematics*, 1(73-77), 1949.

[FG16] Kimon Fountoulakis and Jacek Gondzio. A second-order method for strongly convex ℓ_1-regularization problems. *Mathematical Programming*, 156(1-2):189–219, 2016.

[FR15] Olivier Fercoq and Peter Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.

[FW56] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

[Gab83] Daniel Gabay. Chapter ix applications of the method of multipliers to variational inequalities. *Studies in mathematics and its applications*, 15:299–331, 1983.

[GB16] Pontus Giselsson and Stephen Boyd. Linear convergence and metric selection for Douglas-Rachford Splitting and ADMM. *preprint arXiv:1410.8479*, 2016.

[Gis15] Pontus Giselsson. Tight global linear convergence rate bounds for Douglas-Rachford splitting. *preprint arXiv:1506.01556*, 2015.

[Gon12a] Jacek Gondzio. Interior point methods 25 years later. *European Journal of Operational Research*, 218(3):587–601, 2012.

[Gon12b] Jacek Gondzio. Matrix-free interior point method. *Computational Optimization and Applications*, 51(2):457–480, 2012.

[Gül91] Osman Güler. On the convergence of the proximal point algorithm for convex minimization. *SIAM Journal on Control and Optimization*, 29(2):403–419, 1991.

[GW95] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[HD12] Lauren A Hannah and David B Dunson. Ensemble methods for convex regression with applications to geometric programming based circuit design. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 369–376, 2012.

[HD13] Lauren A Hannah and David B Dunson. Multivariate convex regression with adaptive partitioning. *The Journal of Machine Learning Research*, 14(1):3261–3294, 2013.

[Hil54] Clifford Hildreth. Point estimates of ordinates of concave functions. *Journal of the American Statistical Association*, 49(267):598–619, 1954.

[HJL12] V Härter, Christian Jansson, and Marko Lange. VSDP: A matlab toolbox for verified semidefinite-quadratic-linear programming. Technical report, Technical report, Institute for Reliable Computing, Hamburg University of Technology, 2012, 2012.

[Hub64] Peter J Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

[HUL96] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Convex Analysis and Minimization Algorithms*. Springer Verlag, Heidelberg, 1996. Two volumes - 2nd printing.

[Jag13] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 427–435, 2013.

[JBAS10] Michel Journée, Francis Bach, Pierre-Antoine Absil, and Rodolphe Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010.

[JKJJ08] Bjorn Johansson, Tamás Keviczky, Mikael Johansson, and Karl Henrik Johansson. Subgradient methods and consensus algorithms for solving convex optimization problems. In *Proceedings of the 47th IEEE Conference on Decision and Control (2008)*, pages 4185–4190, 2008.

[KF16a] Donghwan Kim and Jeffrey A Fessler. Another look at the "Fast Iterative Shrinkage/Thresholding Algorithm (FISTA)". *preprint arXiv:1608.03861*, 2016.

[KF16b] Donghwan Kim and Jeffrey A Fessler. Generalizing the optimized gradient method for smooth convex minimization. *preprint arXiv:1607.06764*, 2016.

[KF16c] Donghwan Kim and Jeffrey A Fessler. On the convergence analysis of the optimized gradient method. *Journal of Optimization Theory and Applications*, 2016.

[KF16d] Donghwan Kim and Jeffrey A Fessler. Optimized first-order methods for smooth convex minimization. *Mathematical Programming*, 159(1):81–107, 2016.

[Kiw04] Krzysztof C Kiwiel. Convergence of approximate and incremental subgradient methods for convex optimization. *SIAM Journal on Optimization*, 14(3):807–840, 2004.

[KSST09] Sham Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization, 2009.

[LÖ4] Johan Löfberg. YALMIP : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, 2004.

[LCNS04] Delphine Lambert, Jean-Pierre Crouzeix, V Hien Nguyen, and Jean-Jacques Strodiot. Finite convex integration. *Journal of Convex Analysis*, 11(1):131–146, 2004.

[LM79] Pierre-Louis Lions and Bertrand Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.

[LRP16] Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.

[LV14] Monique Laurent and Antonios Varvitsiotis. A new graph parameter related to bounded rank positive semidefinite matrix completions. *Mathematical Programming*, 145(1-2):291–325, 2014.

[LY08] David G Luenberger and Yinyu Ye. *Linear and nonlinear programming*. Springer, 2008.

[MB09] Alessandro Magnani and Stephen Boyd. Convex piecewise-linear fitting. *Optimization and Engineering*, 10(1):1–17, 2009.

[Mor65] Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.

[Mos10] APS Mosek. The MOSEK optimization software. *Online at http://www.mosek.com*, 54, 2010.

[Nem92] Arkadi S Nemirovski. Information-based complexity of linear operator equations. *Journal of Complexity*, 8(2):153–175, 1992.

[Nem99] Arkadi S Nemirovski. Optimization II: Numerical methods for nonlinear continuous optimization. Lecture notes, 1999. Available from: `http://www2.isye.gatech.edu/~nemirovs/Lect_OptII.pdf`.

[Nes83] Yurii Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$). *Soviet Mathematics Doklady*, 27:372–376, 1983.

[Nes04] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2004.

[Nes05] Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

[Nes08] Yurii Nesterov. How to advance in structural convex optimization. *OPTIMA, MPS Newsletter*, 78:2–5, 2008.

[Nes12a] Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[Nes12b] Yurii Nesterov. How to make the gradients small. *Optima*, 88:10–11, 2012.

[Nes13] Yurii Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[Nes14] Yurii Nesterov. Subgradient methods for huge-scale optimization problems. *Mathematical Programming*, 146(1-2):275–297, 2014.

[NN94] Yurii Nesterov and Arkadi S Nemirovski. *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994.

[NNG15] Ion Necoara, Yurii Nesterov, and François Glineur. Linear convergence of first order methods under weak nondegeneracy assumptions for convex programming. *preprint arXiv:1504.06298*, 2015.

[NO09] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[NOP10] Angelia Nedic, Asuman Ozdaglar, and Pablo A Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, 2010.

[NW06] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[NY83] Arkadi S Nemirovski and David B Yudin. Problem complexity and method efficiency in optimization. *Willey-Interscience, New York*, 1983.

[OHM06] Robert Orsi, Uwe Helmke, and John B Moore. A newton-like method for solving rank constrained linear matrix inequalities. *Automatica*, 42(11):1875–1882, 2006.

[Pat98] Gábor Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Mathematics of Operations Research*, 23(2):339–358, 1998.

[PB13] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):123–231, 2013.

[PE10] Daniel P Palomar and Yonina C Eldar. *Convex optimization in signal processing and communications*. Cambridge university press, 2010.

[Pol64] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[Pol87] Boris T Polyak. *Introduction to Optimization*. Optimization Software New York, 1987.

[PV91] Panos M Pardalos and Stephen A Vavasis. Quadratic programming with one negative eigenvalue is np-hard. *Journal of Global Optimization*, 1(1):15–22, 1991.

[RB16] Ernest Ryu and Stephen Boyd. A primer on monotone operator methods. *Applied and Computational Mathematics an International Journal*, 15(1), 2016.

[Ren01] James Renegar. *A Mathematical View of Interior-Point Methods in Convex Optimization*. SIAM, 2001.

[RM51] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[RM09] James B Rawlings and David Q Mayne. *Model predictive control: Theory and design*. Nob Hill Publishing, 2009.

[Roc76] R Tyrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

[Roc96] R Tyrell Rockafellar. *Convex Analysis*. Princeton University Press, 1996.

[Roc99] R Tyrrell Rockafellar. Second-order convex analysis. *Journal of Nonlinear and Convex Analysis*, 1:1–16, 1999.

[RT14] Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

[Rus06] Andrzej P Ruszczyński. *Nonlinear optimization*. Princeton university press, 2006.

[RW98] R Tyrell Rockafellar and Roger J-B Wets. *Variational Analysis*. Springer, 1998.

[Sah74] Sartaj Sahni. Computationally related problems. *SIAM Journal on Computing*, 3(4):262–279, 1974.

[SBC14] Weijie Su, Stephen Boyd, and Emmanuel Candes. A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems*, pages 2510–2518, 2014.

[SL16] Ziqiang Shi and Rujie Liu. A better convergence analysis of the block coordinate descent method for large scale machine learning. *preprint arXiv:1608.04826*, 2016.

[SLRB11] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Advances in neural information processing systems*, pages 1458–1466, 2011.

[SNW12] Suvrit Sra, Sebastian Nowozin, and Stephen J Wright. *Optimization for machine learning*. Mit Press, 2012.

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.

[ST04] Daniel A Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM (JACM)*, 51(3):385–463, 2004.

[Stu99] Jos F Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

[THG16a] Adrien B Taylor, Julien M Hendrickx, and François Glineur. Smooth strongly convex interpolation and exact worst-case performance of first-order methods. *Mathematical Programming*, 2016.

[THG16b] Adrien B Taylor, Julien M Hendrickx, and François Glineur. Exact worst-case performance of first-order methods for composite convex optimization. *preprint arXiv:1512.07516 (Accepted in SIAM Journal on Optimization)*, 2016.

[THG16c] Adrien B Taylor, Julien M Hendrickx, and François Glineur. Exact and global worst-case convergence of the proximal gradient method for smooth strongly convex optimization. *In preparation*, 2016.

[Tse08] Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. *Submitted to SIAM Journal on Optimization*, 2008.

[VB94] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1994.

[ZC15] Hui Zhang and Lizhi Cheng. Restricted strong convexity and its applications to convergence analysis of gradient-type methods in convex optimization. *Optimization Letters*, 9(5):961–979, 2015.

[ZRM09] Royce KP Zia, Edward F Redish, and Susan R McKay. Making sense of the Legendre transform. *American Journal of Physics*, 77(7):614–622, 2009.