École polytechnique de Louvain

Worst-case functions for the gradient method with fixed variable step sizes

Author: **Diego ELOI**
Supervisor: **François GLINEUR**
Readers: **Julien HENDRICKX, Sébastien COLLA**
Academic year 2021–2022
Master [120] in Mathematical Engineering

Abstract

This master thesis brings new elements in the analysis of the worst-case performances of first-order methods. We have focused this work on the gradient method with fixed variable step sizes for unconstrained smooth (possibly strongly) convex minimization. The aim was to derive the functions that achieve the worst-case behaviors of the gradient method according to objective function accuracy. These behaviors are numerically computed with the PESTO toolbox. PESTO relies on the performance estimation framework and allows the computing of tight worst-case performance of first-order methods by solving a semidefinite program. A conjecture on the exact worst-case bound of two steps of the gradient method for unconstrained smooth (possibly strongly) convex minimization has recently been developed. A conjecture for three steps has also been derived. In this master thesis, we have identified the functions that reach the worst-case bounds conjectured for two and three steps of the gradient method. These functions are called *worst-case functions*. The approach to identify these functions is based on convex interpolation. We also provide some conclusions for N steps of the gradient method.

Keywords: Performance Estimation Problem, Convex Interpolation, Worst-case Function, Gradient Method, Worst-case Bound, Objective Function Accuracy,

Acknowledgement

First of all, I would like to thank my supervisor, Professor François Glineur, for his precious help, his advice and the rich discussions we had on the subject.

I would also like to thank my high school teacher, Mrs. Fievez, for teaching me the math that I love.

Finally, I would like to thank my parents and friends for their support during my years at UCLouvain.

Contents

1 Introduction **3**

2 Elements of convex analysis **5**
2.1 Definitions 5
2.1.1 Convex sets and convex functions 5
2.1.2 Smoothness 7
2.1.3 Strong-Convexity 7
2.2 L-smooth μ-strongly convex interpolation 8
2.3 Conclusion 10

3 Performance estimation problem **11**
3.1 Formal definition 11
3.2 Performance estimation for smooth strongly convex functions 13
3.2.1 Fixed-step first-order methods 14
3.2.2 Formulation as a convex semidefinite program 15
3.3 Homogeneity of the optimal values 17
3.4 PESTO Toolbox 17
3.5 Illustration in an example 17
3.6 Conclusion 19

4 One iteration of gradient method **20**
4.1 Introduction 20
4.2 Smooth convex functions 23
4.3 Smooth strongly convex functions 26
4.4 Conclusion 31

5 Two iterations of gradient method **32**
5.1 Smooth convex functions 32
5.1.1 Region 1 (S,S) 34
5.1.2 Region 2 (B,B) 35
5.1.3 Region 3 (B,S) 37
5.1.4 Region 4 (S,B) 39
5.2 Smooth strongly convex functions 41
5.2.1 Region 1 (S,S) 43
5.2.2 Region 2 (B,B) 46
5.2.3 Region 3 (B,S) 46

5.2.4 Region 4 (S,B) 49
5.3 Conclusion 51

6 Three iterations of gradient method **53**
6.1 Smooth convex functions 53
6.1.1 Region 1 (S,S,S) 54
6.1.2 Region 2 (S,B,B) 55
6.1.3 Region 3 (S,S,B) 57
6.1.4 Region 4 (S,B,S) 58
6.1.5 Region 5 (B,S,S) 60
6.1.6 Region 6 (B,S,B) 62
6.1.7 Region 7 (B,B,B) 63
6.1.8 Region 8 (B,B,S) 65
6.2 Conclusion 65

7 N steps of gradient method **67**
7.1 Smooth convex functions 67
7.2 Smooth strongly convex functions 71
7.3 Conclusion 74

8 Conclusion **75**
8.1 Research outcomes 75
8.2 Further research 77

Bibliography **79**

Chapter 1

Introduction

In industry, it is common to face many problems where one needs to find the best way to accomplish a task, and to choose the best solution from all possible solutions. Often these problems involve searching for a minimum or a maximum of a given function: the minimum cost of production in a factory, the maximum power that can be generated by a device, or the minimum amount of material used to package a product in manufacturing. These are called *Optimization Problems*. In a more mathematical way, the aim of an optimization problem is to find a parameter x within a set $\mathcal{S}$ that gives the smallest value of $f(x)$. We can give a general formulation of such a problem:

$$\min_{x \in \mathcal{S}} f(x) \tag{OP}$$

Almost every problem in daily life can be modelled in the optimization framework. Unfortunately, only a few instances can be solved in practice, at least for large numbers of variables. So optimizers look for methods that guarantee a solution. To do that, they add assumptions on both $\mathcal{S}$ and f. Such assumptions can lead to the design of efficient algorithms, i.e., with working certifications. On the other hand, the purpose is also to design algorithms that can solve a rich class of problems, so these algorithms can be applied to as many applications as possible.

There are many different methods to solve an Optimization Problem (OP). In this master thesis, we will focus our work only on the gradient method (GM), an iterative black-box first-order optimization method. An iterative method is an algorithm that starts with an initial value, and then proceeds iteratively by determining a succession of refined approximate solutions that gradually approach the final solution. However, black-box methods prevent the user from having complete information about the objective function, as they are limited to restricted information at selected points. A first-order method only exploits the values and gradients of points from the domain of f.

Moreover, we will consider only *unconstrained optimization problems*, meaning that $\mathcal{S} = \mathbb{R}^d$. We will also restrict ourselves to the class of smooth convex and smooth strongly convex functions. These functional classes are defined in Chapter 2. We will also establish some properties of such functions and we will end the chapter by introducing the concept of convex interpolation. This notion will be essential for the sequel.

Since there are several first-order methods, it is normal to wonder which one is the best for solving (OP). Which method comes closest to the optimal solution in a fixed number of iterations? In other words, we want to study the performance of an optimization method on a class of problems. We want to know how far a solution given by the algorithm will be from the optimal solution. There are different criteria to characterize the performance of a method. For example we can look at the distance between the values of f at the last iterate and at the solution. We can also look at the difference between the last iterate and the solution, or the norm of the gradient at the last iterate. In this thesis, we will be focus only on the first criterion, the *objective function accuracy*. Obviously, the exact accuracy may vary from one problem to another in the same class. Therefore, we want to derive a *worst-case bound* of our selected performance criterion. This bound corresponds to the performance of the method on the worst problem of the class. The worst-case performances are computed using the PESTO toolbox [THG17a], a Matlab toolbox developed by A. Taylor, J. Hendrickx and F. Glineur. It computes automatically tight worst-case bounds on our selected criterion and for our selected first-order method. All these notions are detailed in Chapter 3.

The main objective of this master thesis is the research and analysis of *worst-case functions* of the gradient method with fixed variable step sizes (worst-case in the sense that they achieve the worst-case bound on the objective function accuracy, when applying the gradient method). Fixed variable step sizes means that the step sizes are fixed in advance, but they are not necessarily equal. We can find in [DT14] and in [Tay17] conjectures for the exact worst-case of the gradient method on our desired performance criterion for fixed constant step sizes. Antoine Daccache derived in his master thesis [Dac19] conjectures for the exact worst-case performance of GM with fixed variable step sizes. Those conjectures are tailored for smooth (possible strongly) convex functions. So the aim of this work was to identify the functions that match the worst-case bounds found in [Dac19]. The results and how we proceeded can be found in Chapters 4,5,6 and 7.

Chapter 2

Elements of convex analysis

As explained in the introduction, we add assumptions on our objective function, in order to get some theoretical guarantees that ensure the good behavior of the algorithms on these problems. Those assumptions rely on the functional classes of interest. In this master thesis, we will study two classes of functions: L-smooth convex and L-smooth μ-strongly convex functions. We will begin this chapter by giving some definitions and properties of such classes of functions. At the end of this chapter, we will define the concept of convex interpolation. It is an important concept, which we use later in this thesis. Our main references for this chapter are [Tay17] and [Nes18].

Also, we consider only an unconstrained optimization problem. Therefore the set of interest is $\mathcal{S} = \mathbb{R}^d$. Thus, we will restrict ourselves to the real d-space $\mathbb{R}^d$ and we will use the standard inner product $\langle x, y \rangle = x^T y \; \forall x, y \in \mathbb{R}^d$ and the Euclidean norm $\|x\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_d^2} = \sqrt{x^T x}$.

2.1 Definitions

2.1.1 Convex sets and convex functions

We start by recalling the definition of a convex set. An intuitive view of this notion is that a line segment joining any two points of that set lies completely within the set.

Definition 2.1. A set $Q \subseteq \mathbb{R}^d$ is convex if and only if for any $x, y \in Q$ and for any $\lambda \in [0, 1]$:

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

For the following, we consider non-empty closed convex sets. Non-closed sets may turn simple problems into ill-defined ones, i.e., having no solution. Such problems are often very impractical to work with. Since we consider only an unconstrained optimization problem, the set of interest will be $Q = \mathbb{R}^d$. We can also define differently the convexity of a set, if it is closed. This definition relies on the use of supporting closed half-spaces. We then state the following theorem from [Tay17].

Theorem 2.2. ([Tay17, Theorem 2.4]) Consider $Q \subseteq \mathbb{R}^d$ a closed set with a non-empty interior. We say that Q is convex if and only if for every point x_0 of its boundary, there exists an hyperplane $\{x \in \mathbb{R}^d | \langle a, x \rangle = b\}$ such that $\langle a, x \rangle \leq b \; \forall x \in Q$ and $\langle a, x_0 \rangle = b$.

Now that we have defined a convex set, we can look at convex functions. The convexity of functions has interesting properties, especially in optimization theory. We will see at the end of this section how we can take advantage of such functions. We start by defining the epigraph of a function.

Definition 2.3. Consider a function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$. We denote by epi f the epigraph of f defined as follows:

$$\text{epi } f = \{(x, t) \in \mathbb{R}^d \times \mathbb{R} : f(x) \leq t\}$$

The first way to define convex functions is to take the definition of a convex set and apply it to the epigraph of the function. This leads to the following definition.

Definition 2.4. Consider a function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$. Function f is convex if and only if its epigraph is a convex set.

As for convex set, we restrict ourselves to closed proper functions, meaning that their epigraph is a non-empty closed set.

Notations 2.5. We denote by $\mathcal{F}_{0,\infty}(\mathbb{R}^d)$ the class of convex closed proper functions.

We also state a general definition of convex functions, which works with or without differentiability.

Definition 2.6. A function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is convex if $\forall x, y \in \mathbb{R}^d$ and for any $\lambda \in [0, 1]$:

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

Graphically, this means that the line segment connecting each pair of points $(x, f(x))$ and $(y, f(y))$ must sit above the graph of f. As we did for convex sets, we can use supporting hyperplanes applied to the epigraph to define the convexity of a function. These hyperplanes are usually non-vertical and are in a one-to-one correspondence with the subgradients. The latter is defined as follows:

Definition 2.7. Let $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ be a function. A vector g is called a subgradient of function f at point $x_0 \in \operatorname{dom} f$ if for any $x \in \operatorname{dom} f$ we have:

$$f(x) \geq f(x_0) + \langle g, x - x_0 \rangle$$

Therefore, we can geometrically interpret subgradients as hyperplanes supporting the epigraph of f at $(x, f(x))$. Subgradients also correspond to global under-estimators of f. The set of all subgradients of f at x_0 is denoted by $\partial f(x_0)$ and is called the subdifferential of function f at the point x_0. Also note that subdifferentiability everywhere of a function implies convexity. When dealing with differentiable functions, the subdifferential at each point reduces to a singleton containing the gradient. In other words, the subgradient replaces the gradient for non-differentiable functions. For differentiable functions, we have the following definition of a convex function:

Definition 2.8. A differentiable function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is convex if $\forall x, y \in \mathbb{R}^d$:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

The theorem below states the link between convex functions and the notion of subdifferential. We refer to [Nes18, Theorem 3.1.13] for the proof.

Theorem 2.9. Consider a function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$. Function f is convex if and only if $\forall \in \text{int}(\text{dom } f)$, the set $\partial f(x)$ is non-empty.

As explained previously, we can take advantage of working with convex functions in the field of optimization. The following theorem states that every local optimum is global for convex functions.

Theorem 2.10. Let $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ be a convex and proper function, then $0 \in \partial f(x_*)$ if and only if $f(x_*) = \underset{x \in Dom f}{min} f(x)$.

Proof: If $0 \in \partial f(x_*)$, then we have $f(x) \geq f(x_*) + \langle 0, x - x_* \rangle = f(x_*)$ valid $\forall x \in \text{dom } f$. If $f(x_*) = \underset{x \in Dom f}{min} f(x)$ we have that $0 \in \partial f(x_*)$ directly from Definition (2.7.).

2.1.2 Smoothness

In this section, we describe the notion of smoothness. As well explained in [Nes18, Section 1.2.2], there are no interesting properties on the minimization methods when one only admits differentiability. To compensate for this, we add some conditions (i.e., smoothness conditions) on the magnitude of the derivatives of the functions of interest. We begin by giving a definition of smoothness of closed proper convex differentiable functions.

Definition 2.11. Consider $f \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ and a constant $L \in \mathbb{R}^+$. We say that f is L-smooth if it satisfies:

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2.$$

This is the well-known definition of L-smooth functions with a Lipschitz continuous gradient with constant L.

Notations 2.12. We denote by $\mathcal{F}_{0,L}(\mathbb{R}^d)$ the class of L-smooth convex closed proper functions.

Another way to define smoothness is to require the function to be upper bounded by its first-order development plus a quadratic term. Let's state this in the following theorem:

Theorem 2.13. Consider $f \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$. We have $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$ if and only if $\forall x, y \in \mathbb{R}^d$ we have:

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|x - y\|_2^2 \tag{2.1}$$

2.1.3 Strong-Convexity

The last assumption on the objective function that we will use is strong convexity. As for convex functions, we begin with a very general definition that works also without differentiability.

Definition 2.14. A closed proper convex function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is said to be μ-strongly convex with $\mu \in \mathbb{R}^+$ if for any $x, y \in \mathbb{R}^d$ and for any $\lambda \in [0, 1]$:

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) - \lambda(1 - \lambda)\frac{\mu}{2}\|x - y\|_2^2$$

Note that when $\mu = 0$ we fall back on Definition (2.6.) and therefore the function will be only convex.

Notations 2.15. We denote by $\mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$ the class of μ-strongly convex closed proper functions.

The following theorem characterizes the strong convexity of a function in a different way. It corresponds to [Tay17, Theorem 2.31].

Theorem 2.16. $f \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$ if and only if $f(x) - \frac{\mu}{2}\|x\|_2^2 \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$.

A last equivalent way to view the strong convexity character of functions is given by the following:

Theorem 2.17. ([Tay17, Theorem 2.32]). A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is called μ-strongly convex if there exists a constant $\mu \geq 0$ such that for any $x, y \in \mathbb{R}^d$ we have:

$$f(y) \geq f(x) + \nabla f(x)^T(y - x) + \frac{1}{2}\mu\|y - x\|_2^2.$$

We finish this section by defining the class of L-smooth μ-strongly convex closed proper functions.

Notations 2.18. The class of L-smooth μ-strongly convex closed proper functions with $L \in \mathbb{R} \cup \{\infty\}$ and $\mu \in \mathbb{R}^+$ is denoted by $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$.

Using the previous theorem, we can add the smoothness to the strong convexity property as explained in our main reference [Tay17].

Theorem 2.19. Consider a function $f \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$. We have that $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ if and only if $f(x) - \frac{\mu}{2}\|x\|_2^2 \in \mathcal{F}_{0,L-\mu}(\mathbb{R}^d)$.

2.2 L-smooth μ-strongly convex interpolation

For the sequel of this chapter we will consider a set of points $S = \{x_i, f_i, g_i\}_{i \in I}$ where I denotes a set of indices. The objective of this final section is to establish necessary and sufficient conditions for the set S to be interpolable by a function $F \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$. As a reminder, $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$ denotes the class of L-smooth μ-strongly convex closed proper functions. We begin our explanations with the following definition:

Definition 2.20. Let I be a finite index set and consider the set of points $S = \{x_i, f_i, g_i\}_{i \in I}$ with $x_i, g_i \in \mathbb{R}^d$ and $f_i \in \mathbb{R}$. The set S is $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$-interpolable if and only if there exist a function $F \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ such that for all $i \in I$ we have $g_i \in \partial F(x_i)$ and $F(x_i) = f_i$.

To simplify, we will start by finding necessary and sufficient conditions for smooth convex interpolation. A naive approach is to discretize the main inequalities of the class with the points within the set S. For example, we know from the previous sections that the class of L-smooth convex functions is characterized by the two following inequalities

$$\begin{aligned} &\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2, \quad \forall x, y \in \mathbb{R}^d \\ &f(x) \geq f\,(y) + \nabla f(y)^T(x - y), \quad \forall x, y \in \mathbb{R}^d \end{aligned} \tag{2.2}$$

Then, we restrict those conditions to the points of the set S. In other words, the Conditions (2.2) need to be respected for all the pairs x_i, x_j with $i, j \in I$. The Conditions (2.2) then becomes

$$\begin{aligned} |g_i - g_j| \leq L|x_i - x_j|, \quad \forall i, j \in I \\ f_i \geq f_j + g_j^T(x_i - x_j), \quad \forall i, j \in I \end{aligned} \tag{2.3}$$

However this naive approach is wrong. Because although those conditions are necessary for smooth convex interpolation, they are not sufficient. Even though the Conditions (2.2) are sufficient to guarantee that $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$, they are not sufficient for the set S to be interpolated. Let's take an example to show this.

Example 2.1. In this example, $I = \{1, 2\}$ and the two triples are given by

$$(x_1, f_1, g_1) = (0, 3, -1) \quad (x_2, f_2, g_2) = (3, \frac{3}{2}, -\frac{1}{2})$$

We want to interpolate these points by a smooth convex function $F \in \mathcal{F}_{0,L}(\mathbb{R}^d)$. The Conditions (2.3) are satisfied for $L = 1$. Unfortunately, for any finite value of L, we can't find a smooth convex function that interpolates the set S. The interpolating function must lie entirely above its linear under-approximations due to the convexity requirement. That implies non-differentiability at x_1 as we can see in Figure 2.1.

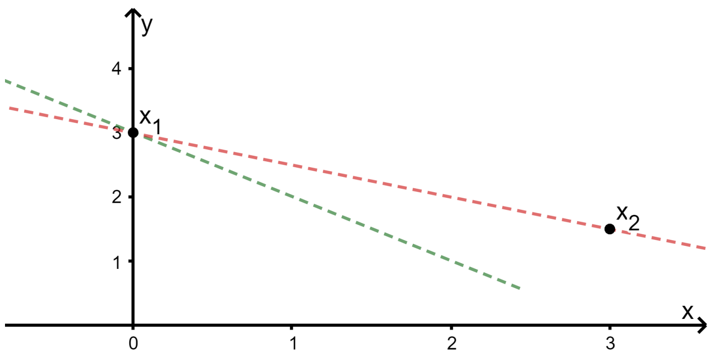


Figure 2.1: Interpolation of the set S. The red line segment has a slope of $-\frac{1}{2}$ and the green line segment has a slope of -1.

Hence the naive approach, which involves discretizing conditions that are necessary and sufficient on the whole space shows some weakness. If we face a constraint of the form $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$ in an optimization problem and we replace it by the Conditions (2.3), we could obtain an interpolating function that does not belong to the desired class function. However this example shows us that the convexity constraint in Conditions (2.3) are necessary and sufficient to ensure convex interpolation. We describe these interpolation conditions more formally in the following theorem:

Theorem 2.21. The set of points $S = \{x_i, f_i, g_i\}_{i \in I}$ is $\mathcal{F}_{0,\infty}(\mathbb{R}^d)$-interpolable if and only if

$$f_i \geq f_j + g_j^T(x_i - x_j), \quad \forall i, j \in I$$

We refer to [Tay17, Theorem 3.4] for a proof. Finally we state in broad terms the necessary and sufficient conditions of smooth strongly convex interpolation. They are the main conditions that we will face in the sequel of this work. More details and proofs can be found in [Tay17].

Theorem 2.22. The set of points $S = \{x_i, f_i, g_i\}_{i \in I}$ is $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$-interpolable if and only if the following inequalities hold for every pair $(i, j) \in I^2$

$$\begin{aligned} f_i \geq f_j + g_j^T \left(x_i - x_j\right) + \frac{1}{2 - \mu/L} \Bigg(\frac{1}{L} \left\|g_i - g_j\right\|_2^2 \\ + \mu \left\|x_i - x_j\right\|_2^2 - 2\frac{\mu}{L} \left(g_i - g_j\right)^T \left(x_j - x_i\right) \Bigg) \end{aligned} \tag{2.4}$$

Finally, we can derive the interpolation conditions to be met for the smooth convex case, i.e., when $\mu = 0$.

Corollary 2.23. The set of points $S = \{x_i, f_i, g_i\}_{i \in I}$ is $\mathcal{F}_{0,L}(\mathbb{R}^d)$-interpolable if and only if the following inequalities hold for every pair $(i, j) \in I^2$

$$f_i \geq f_j + g_j^T \left(x_i - x_j\right) + \frac{1}{2L} \left\|g_i - g_j\right\|_2^2$$

Taking the pair of triples from previous example, the condition of Corollary (2.23.) becomes:

$$0 \geq \frac{9}{8L}$$

Since $L \in \mathbb{R}^+$, the condition can not be respected. Therefore, the pair of triples is not $\mathcal{F}_{0,L}(\mathbb{R}^d)$-interpolable.

2.3 Conclusion

In this chapter, we explained some definitions and theorems about the class of functions mainly used in the field of optimization. Our classes of interest for the sequel of this master thesis will be the classes of smooth strongly convex and smooth convex functions, denoted by $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$ and $\mathcal{F}_{0,L}(\mathbb{R}^d)$ respectively. We also set the stage for the next chapter by defining the necessary and sufficient conditions to ensure that a set of points is $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$-interpolable or $\mathcal{F}_{0,L}(\mathbb{R}^d)$-interpolable. These notions will be the basis to formulate our main problem and for finding the worst-case functions that are the purpose of this work.

Chapter 3

Performance estimation problem

In this third chapter we describe the performance estimation problems (PEPs) based on the PhD thesis of Adrien Taylor [Tay17]. This will be our main reference again, together with [THG17c] and [THG17b]. The performance estimation problems were initially introduced in the works of Drori and Teboulle [DT14], who introduced a novel approach for analyzing the performance of first-order black-box optimization methods. They formulate the worst-case behavior of such methods in terms of the absolute objective inaccuracy as an optimization problem. Their work was also focused only on smooth unconstrained convex minimization. At the end, they were able to compute upper bounds on the worst-case performance by solving relaxations of (PEPs). Later, A. Taylor, J. Hendrickx and F. Glineur [THG17c] proposed a generic way for formulating and solving the performance estimation problems, in order to achieve guaranteed tight results.

The main purpose of this chapter is to write the performance estimation problem as a convex semidefinite program that can be solved numerically and provides tight worst-case bounds. These bounds are numerically solved with the PESTO toolbox [THG17a], which was developed by A. Taylor during his thesis [Tay17]. We will end the chapter with a simple example that illustrates all the concepts discussed.

3.1 Formal definition

In this chapter, we will consider the following unconstrained minimization problem

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

where f belongs to the class of L-smooth convex or L-smooth μ-strongly convex function. We denote the considered class by $\mathcal{F}$. As explained in the introduction, we will solve this problem using a first-order black-box method. As a reminder, this method relies on the computation of f and its gradient at a sequence of iterates. In other words, the method $\mathcal{M}$ does not have access to all the information on the considered function f. To have access to the function values and the gradients, we provide the method with a first order oracle $\mathcal{O}_f(x) = \{f(x), \nabla f(x)\}$. Consider N iterates generated by a first-order black-box method $\mathcal{M}$, starting from an initial point x_0.

We have the following relationship between them:

$$\begin{aligned} x_1 &= \mathcal{M}_1\left(x_0, \mathcal{O}_f\left(x_0\right)\right) \\ x_2 &= \mathcal{M}_2\left(x_0, \mathcal{O}_f\left(x_0\right), \mathcal{O}_f\left(x_1\right)\right) \\ &\vdots \\ x_N &= \mathcal{M}_N\left(x_0, \mathcal{O}_f\left(x_0\right), \ldots, \mathcal{O}_f\left(x_{N-1}\right)\right) \end{aligned} \tag{3.1}$$

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

Our objective is to study the accuracy of the method on a certain class of problems. Of course, the accuracy is not constant from one problem to another in the same class. So we will derive a worst-case bound on the accuracy. The bound corresponds to the performance of the method on the worst problem of the class. We can study the accuracy on many possible performance criteria. For example, the function value accuracy $f(x_N) - f(x_*)$, the squared residual gradient norm $\|\nabla f(x_N)\|_2^2$ and the squared distance to an optimal solution $\|x_N - x_*\|_2^2$. We denote by x_* any minimizer of f. This additional point is obtained thanks to the oracle. Given the type of method we use, these criteria can only be computed with the information we get from the first-order oracle. We will focus our work on the objective function value accuracy and we denote the performance criteria by $\mathcal{P}$. The worst-case performance is then obtained by maximizing the performance criterion $\mathcal{P}$ over the functions in the class $\mathcal{F}$.

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

The parameter R is settled to bound the distance from the initial iterate x_0 and the optimum x_*. In most situations, the performance of a first-order method cannot be sensibly assessed without such a constraint.

Evidently, the problem (PEP) involves an unknown function as a variable. Our problem is then infinite dimensional. To avoid that, we can use the black-box method property. Indeed, only the values given by the first-order oracle are needed. We can define the set $I = \{0, 1, \ldots, N, *\}$ corresponding to the indices of the iterates and we denote the output of the oracle at each iterate by $\mathcal{O}_f(x_i) = \{f_i, g_i\}$. We can then reformulate the problem (PEP) into a finite-dimensional optimization problem using only the iterates $\{x_i\}_{i\in I}$, their function values $\{f_i\}_{i\in I}$ and their gradients $\{g_i\}_{i\in I}$:

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

The first constraint requires that the function f belonging to the class $\mathcal{F}$ must interpolate the set of variables $\{x_i,g_i,f_i\}$. Any solution of the problem (PEP) can be discretized to provide a solution to the problem (f-PEP). Inversely, any solution of the problem (f-PEP) can be interpolated to provide a solution to (PEP). So it is clear that the optimal values of the two problems are equal and we get:

$$w^f(\mathcal{M},\mathcal{F},R,\mathcal{P},N) = w(\mathcal{M},\mathcal{F},R,\mathcal{P},N).$$

3.2 Performance estimation for smooth strongly convex functions

In this section we restrict ourselves to the class of smooth strongly convex functions defined on $\mathbb{R}^d$. As a reminder, this class is denoted by $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$. We then have an instance of the problem (f-PEP). This specific class of functions and the considered first-order method are invariant with respect to translation in their domain and additive shifts in the function values. To simplify what is coming next, we assume without loss of generality that $x_*=0$ and $f_*=0$. This particular case of (f-PEP) can be formulated as follows:

$$\begin{aligned} w^d_{\mu,L}(\mathcal{M},R,\mathcal{P},N) = \sup_{\{x_i,f_i,g_i\}_{i\in I}\in\left(\mathbb{R}^d\times\mathbb{R}^d\times\mathbb{R}\right)^{N+2}} \quad & \mathcal{P}\left(\{x_i,f_i,g_i\}_{i\in I}\right) \\ \text{such that } \{x_i,g_i,f_i\}_{i\in I} \text{ is } & \mathcal{F}_{\mu,L}-\text{interpolable}, \\ & x_1,\dots,x_N \text{ is generated from } x_0 \text{ by method} \\ & \quad \mathcal{M} \text{ using } \{f_i,g_i\}_{i\in\{0,\dots,N-1\}}, \\ & \{x_*,g_*,f_*\}=\{0,0,0\}, \\ & \|x_0-x_*\|_2\le R. \end{aligned} \tag{d-PEP}$$

We can easily deduce that $w^d_{\mu,L}(\mathcal{M},R,\mathcal{P},N) = w(\mathcal{M},\mathcal{F}_{\mu,L}(\mathbb{R}^d),R,\mathcal{P},N)$ since (d-PEP) is an instance of (f-PEP). The second constraint in (d-PEP) means that the set of points $\{x_i,g_i,f_i\}_{i\in I}$ must respect the interpolation conditions given by Theorem (2.22.). However, because of this constraint and especially the term $g_j^T(x_i-x_j)$, the problem (d-PEP) is not convex. We will see in the next subsection how to transform this problem into a convex semidefinite program. This can be done when dealing with fixed-step first-order black-box methods.

3.2.1 Fixed-step first-order methods

We limit ourselves to the class of fixed-step first-order methods. Fixed-steps means that the steps are fixed in advance. Here is a definition:

Definition 3.1. A method $\mathcal{M}$ is a fixed-step method if its iterates are computed according to

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

where $h_{i,k}$ is a scalar coefficient.

Such a method applying N steps is represented by the lower triangular matrix $H = \{h_{i,k}\}_{1\leq i\leq N, 0\leq k\leq N-1} \in \mathbb{R}^{N\times N}$ with $h_{i,k} = 0$ if $k \geq i$.

Example 3.1. The gradient method with fixed variable step sizes that we will use later is included in this class of method. Indeed by setting $h_{i,k} = \eta_{k+1}$ for $0 \leq k \leq i-1$, we can write the method as follows

$$x_i = x_0 - \sum_{k=0}^{i-1} \eta_{k+1} g_k$$

Where the matrix H is given by

$$H = \frac{1}{L}\begin{bmatrix} \eta_1 & 0 & \cdots & 0 \\ \eta_1 & \eta_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \eta_1 & \eta_2 & \cdots & \eta_N \end{bmatrix}$$

For example, the 4^{th} iterate is defined as follows

$$x_4 = x_0 - \frac{1}{L}(\eta_1 g_0 + \eta_2 g_1 + \eta_3 g_2 + \eta_4 g_3)$$

Let's continue our reformulation. Firstly, we define the vector P as follows:

$$P = [g_0, g_1, \ldots, g_N, x_0]$$

Then we define a symmetric Gram matrix $G = P^T P \in \mathbb{S}^{N+2}$ ($\mathbb{S}^d$ denote the set of symmetric matrices of size $d \times d$), which is equivalent to

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

3.2.2 Formulation as a convex semidefinite program

Let us now see the usefulness of the Gram matrix G. All iterates of (d-PEP) can be obtained using the Definition (3.1.) of fixed-step first-order methods and the resulting formulation only involves function values f_i and inner products between x_0 and all gradients. Consequently, starting from x_0, all the constraints in problem (d-PEP) can be entirely formulated in terms of the entries of G and the function values f_i. Furthermore, the Gram matrix G should be semidefinite positive, i.e., $G \succeq 0$. Finally, the rank of G is at most d since vectors x_0 and g_i belong to $\mathbb{R}^d$.

From now on, the objective is to rewrite the interpolation conditions of Theorem (2.22.) using the Gram matrix G and the function values f_i. To do that, let's begin with some definitions.

- We define vectors $h_i \in \mathbb{R}^{N+2}\ \forall i \in \{0, \dots, N\}$ and $h_* \in \mathbb{R}^{N+2}$ as follows

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

  in order to have $x_i = P h_i$ (see Definition (3.1.)).

- We also define $u_i = e_{i+1} \in \mathbb{R}^{N+2}$, the canonical basis vectors.

- Finally we define u_* as the vector of zeros.

We can now rewrite the interpolation constraint of (d-PEP) for all $i, j \in I$ as follows:

$$\begin{aligned} f_i \geq f_j &+ \frac{L}{L-\mu}\left(u_j^\top G h_i - u_j^\top G h_j\right) + \frac{1}{2(L-\mu)}\left(u_i - u_j\right)^\top G \left(u_i - u_j\right) \\ &+ \frac{\mu}{L-\mu}\left(u_i^\top G h_j - u_i^\top G h_i\right) + \frac{L\mu}{2(L-\mu)}\left(h_i - h_j\right)^\top G \left(h_i - h_j\right) \end{aligned} \tag{3.2}$$

It is also possible to formulate these constraints as well as the initial condition $\|x_0 - x_*\|_2 \leq R$ using the trace operator. In order to do so, we define the matrices A_{ij} and A_R for all $i, j \in I$ as follows:

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

Then we can rewrite all the constraints of problem (d-PEP) in the following compact form:

$$\begin{aligned} f_j - f_i + \mathrm{Tr}\left(GA_{ij}\right) &\leq 0 \quad \forall\, i, j \in I \\ \mathrm{Tr}\left(GA_R\right) - R^2 &\leq 0 \\ G &\succeq 0 \end{aligned} \tag{3.3}$$

We add the following non-convex rank constraint to impose the dimension of the original problem

$$\operatorname{rank} G \leq d$$

The set of constraints (3.3) are linear in their variables $f \in \mathbb{R}^{N+1}$ and $G \in \mathbb{S}^{N+2}$. Finally, we consider the performance criteria $\mathcal{P}$. We observe that if $\mathcal{P}$ is a concave semidefinite-representable function in G and f then the worst-case estimation problem can be cast into a convex semidefinite optimization problem plus a rank constraint. For example, linear functions in G and f are suitable and our three performance criteria that we have already presented are part of it.

We finally attained the main objective of this chapter, i.e., casting our performance estimation problem into a convex semidefinite program. This program is provided in the following theorem. We refer to [Tay17, Theorem 4.2] for a proof.

Theorem 3.2. Consider the class of L-smooth μ-strongly convex functions $\mathcal{F}_{\mu,Ł}(\mathbb{R}^d)$, a fixed-step first-order method that computes N iterates according to matrix $H \in \mathbb{R}^{N\times N}$, and a performance criterion $\mathcal{P}$ that depends linearly on the function values at the iterates and quadratically on these iterates and their gradients. This criterion is defined by $\mathcal{P}_{b,C}(f,G) = b^T f + \text{Tr}(CG)$ with $b \in \mathbb{R}^{N+1}$ and $C \in \mathbb{S}^{N+2}$. The worst-case performance after N iterations of the method $\mathcal{M}$ is obtained by resolving the following rank-constrained semidefinite program:

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

We can observe that the Gram matrix G has a rank at most equal to $N+2$ since $G \in \mathbb{S}^{N+2}$. Therefore, if $N \leq d-2$, the worst-case performance is given by the following convex semidefinite program:

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

In that case, when $N+2 \leq d$, we have an equality between the optimal values:

$$w^{sdp}_{\mu,L}(R,H,N,b,C) = w^{sdp(d)}_{\mu,L}(R,H,N,b,C) \quad \forall\, d \geq N+2 \tag{3.4}$$

As we can see, this last formulation does not depend on the dimension value d. It means that we are able to compute worst-case performances independently of d provided that the functional class of interest contains functions of dimension at least $N+2$.

When we are dealing with the formulation (sdp-PEP(d)), we can simply establish that the sequence $\{w^{sdp(d)}_{\mu,L}(R,H,N,b,C)\}_{d=1,2,\dots}$ is monotonically increasing with respect to the dimension d. In addition to this, from [Tay17, Corollary 4.5] we know that problem (sdp-PEP) guarantees that its optimal value can be achieved by an $(N+2)$-dimensional L-smooth μ-strongly convex function. By combining these two statements with equality (3.4) we can show that $w_{\mu,L}^{sdp(d)}$ increases until at most $d = N+2$. Indeed, the sequence $\{w_{\mu,L}^{sdp(d)}(R,H,N,b,C)\}_{d=1,2,\ldots}$ converges for a finite value of d. Consequently, if we obtain a function of dimension $d_* \leq N+2$ when solving (sdp-PEP), then the resulting optimal value $w_{\mu,L}^{sdp}$ is equal to the exact worst-case performance for all functions of dimensions greater or equal than d_*. For example, on the basis of the formulation (sdp-PEP) for the gradient method, there are numerical validations that provide us with one-dimensional worst-case functions. So they are also the solutions to the problem (sdp-PEP(d)) for any value of d greater than one.

3.3 Homogeneity of the optimal values

In what follows, we will only solve numerically our sdp-PEP problems with the parameters R and L set to one. Indeed, we can deduce from those problems a general valid bound for every value of L and R. As explained in [Tay17, section 4.2.5], for our desired performance criterion, the function accuracy, we have the following homogeneity relation of the optimal values of sdp-PEP with respect to L and R:

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

where $\kappa = \frac{\mu}{L}$ is the inverse condition number and $\frac{H}{L}$ is the representation of the fixed-step method where each step size $h_{i,j}$ is divided by the Lipschitz constant L (i.e., is normalized). For the worst-case functions we will also first deduce them in the case $L = R = 1$ and then by appropriate scaling we will deduce the worst-case functions valid for any value of these parameters.

3.4 PESTO Toolbox

To compute these worst-case bounds, we will use a toolbox called PESTO [THG17a], which is implemented in Matlab and allows users to compute automatically tight worst-case performance for a wide class of first-order optimization methods. The toolbox is based on the use of YALMIP [Löf04], a modeling language as well as the use of an SDP solver such as MOSEK [Mos10] or Sedumi [Stu99]. Its purpose is to facilitate the use of the performance estimation methodology, which we described previously, by allowing users to write the algorithms in a normal way. PESTO then takes care of the modeling and worst-case analysis parts. More details and explanation on how to handle the toolbox can be found in [THG17a].

In the next chapter, we will try to derive the worst-case functions associated with the objective function accuracy of the gradient method. The PESTO toolbox will be useful because it provides us with the set of triples $\{x_i, g_i, f_i\}_{i\in I}$ (with $I = \{0, 1, \ldots, N, *\}$) corresponding to each performance estimation problem that we will solve.

3.5 Illustration in an example

We will illustrate all the notions and concept discussed in this chapter with a simple example. We therefore consider a method that performs a single gradient step. Let's denote this method by $\mathcal{M}$. The normalized step size used is equal to $1/L$. Our single iterate is $x_1 = x_0 - \frac{1}{L}\nabla f(x_0)$. We consider that function f is L-smooth convex, i.e., $f \in \mathcal{F}_{0,L}$. The purpose is to compute the worst-case bound on the objective function accuracy after performing the step. The performance estimation problem (PEP) can be written as follows

$$\begin{aligned} w(\mathcal{M}, \mathcal{F}_{0,L}, R, f(x_1) - f(x_*), 1) = \sup_{f, x_0, x_1, x_*} \; & f(x_1) - f(x_*) \\ \text{such that } & f \in \mathcal{F}_{0,L} \\ & x_* \text{ is optimal for } f, \\ & x_1 = x_0 - \frac{1}{L}\nabla f(x_0), \\ & \|x_0 - x_*\|_2 \leq R. \end{aligned}$$

As stated previously, this problem is infinite dimensional. In order to obtain a finite-dimensional problem, we can write our problem using the formulation (f-PEP), i.e., using only the iterates x_0, x_1, their functions values f_1, f_2 and their gradients g_1, g_2.

$$\begin{aligned} w^f(\mathcal{M}, \mathcal{F}_{0,L}, R, f_1 - f_*, 1) = \sup_{\{x_i, g_i, f_i\}_{i \in \{0,1,*\}}} \; & f_1 - f_* \\ \text{such that there exists } f \in \mathcal{F}_{0,L} \text{ such that } & \mathcal{O}_f(x_i) = \{f_i, g_i\} \; \forall i \in \{0, 1, *\}, \\ & g_* = 0, \\ & x_1 = x_0 - \frac{1}{L} g_0 \\ & \|x_0 - x_*\|_2 \leq R. \end{aligned}$$

We can assume that $x_* = 0$ and $f_* = 0$ without loss of generality and based on the formulation (d-PEP) we can rewrite our problem as

$$\begin{aligned} w^d_{0,L}(\mathcal{M}, R, f_1 - f_*, 1) = \sup_{\{x_i, f_i, g_i\}_{i \in \{0,1,*\}} \in \left(\mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}\right)^{N+2}} \; & f_1 - f_* \\ \text{such that } & \{x_i, g_i, f_i\}_{i \in \{0,1\}} \text{ is } \mathcal{F}_{0,L} - \text{interpolable}, \\ & x_1 = x_0 - \frac{1}{L} g_0 \\ & \{x_*, g_*, f_*\} = \{0, 0, 0\}, \\ & \|x_0 - x_*\|_2 \leq R. \end{aligned}$$

Finally, we can transform our problem into a convex semidefinite program. Since the method $\mathcal{M}$ is a fixed-step first-order method, this transformation is possible. Therefore, the corresponding formulation (sdp-PEP) with $N = 1$, $H = 1$, $\mu = 0$, $C = 0^{3\times 3}$, $b = (0 \;\; 1)^\top$ is written as follows

$$\begin{aligned} w^{sdp}_{0,L}(R, 1, 1, (0 \;\; 1)^\top, 0^{3\times 3}) = \sup_{f \in \mathbb{R}^2, G \in \mathbb{S}^3} \; & f_1 \\ \text{such that } & f_j - f_i + \operatorname{Tr}\left(G A_{ij}\right) \leq 0 \quad \forall\, i, j \in \{0, 1\} \\ & \operatorname{Tr}\left(G A_R\right) - R^2 \leq 0 \\ & G \succeq 0 \end{aligned}$$

where the Gram Matrix is given by

$$G = \begin{pmatrix} g_0^\top g_0 & g_0^\top g_1 & g_0^\top x_0 \\ g_1^\top g_0 & g_1^\top g_1 & g_1^\top x_0 \\ x_0^\top g_0 & x_0^\top g_1 & x_0^\top x_0 \end{pmatrix} \succeq 0$$

Solving this problem with the PESTO toolbox [THG17a] with L and R set to one gives us the following solution

$$w_{0,L}^{sdp}\left(R, 1, 1, (0\ \ 1)^\top, 0^{3\times 3}\right) = \frac{LR^2}{6} \tag{3.5}$$

We used the homogeneity of the optimal values with respect to L and R to generalize the optimal solution to any value of these parameters. The optimal Gram matrix is now given by

$$G = LR^2 \begin{pmatrix} L/9 & L/9 & 1/3 \\ L/9 & L/9 & 1/3 \\ 1/3 & 1/3 & 1/L \end{pmatrix} \succeq 0$$

It is clear that the matrix G is of rank one. So the function that achieves the worst-case bound (3.5) is one-dimensional. We will see in the next chapter how we can find this function and if there is only one worst-case function. Finally, $f(x_1) - f(x_*) \leq \frac{LR^2}{6}$ holds for any function $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$ for any value of d, for the same matrix H and provided that $\|x_0 - x_*\|_2 \leq R$.

3.6 Conclusion

In this third chapter, we introduced the performance estimation problem, i.e., the search for the exact worst-case bound on some performance criteria of black-box first-order optimization methods. We firstly formulated this problem as an optimization problem (PEP) over a set of functions belonging to a certain class $\mathcal{F}$. Then using the property of black-box methods, we write our (PEP) problem as a finite-dimensional optimization problem (f-PEP). The next step was to restrict ourselves to the class of smooth strongly convex functions $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$ and using the notion of convex interpolation we introduced a new formulation (d-PEP). Finally we were able to express our problem as a convex semidefinite program (sdp-PEP), which can be solved numerically. Note that this latter formulation is only valid for fixed-step first-order methods. In the next chapter, the method of interest will be the gradient method with fixed variable step sizes. The purpose will be to derive the worst-case functions according to the objective function accuracy.

Chapter 4

One iteration of gradient method

4.1 Introduction

In this chapter, we will derive the worst-case functions for one iteration of the gradient method equipped with the objective function accuracy criterion and with variable normalized step sizes. From the results obtained for one step, we will deduce in the next chapter the worst-case functions for two steps of the gradient method. We begin by giving some well-known results on the performance of the gradient method with fixed constant step sizes. Firstly, here again is the method.

Gradient Method (GM) with fixed constant step sizes

Input: $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, $x_0 \in \mathbb{R}^d$, $0 \leq h \leq 2$
For $i = 1, \ldots, N$

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

As a reminder, $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$ denotes the class of L-smooth μ-strongly convex functions. We can also find in [DT14] the following conjecture for smooth convex functions.

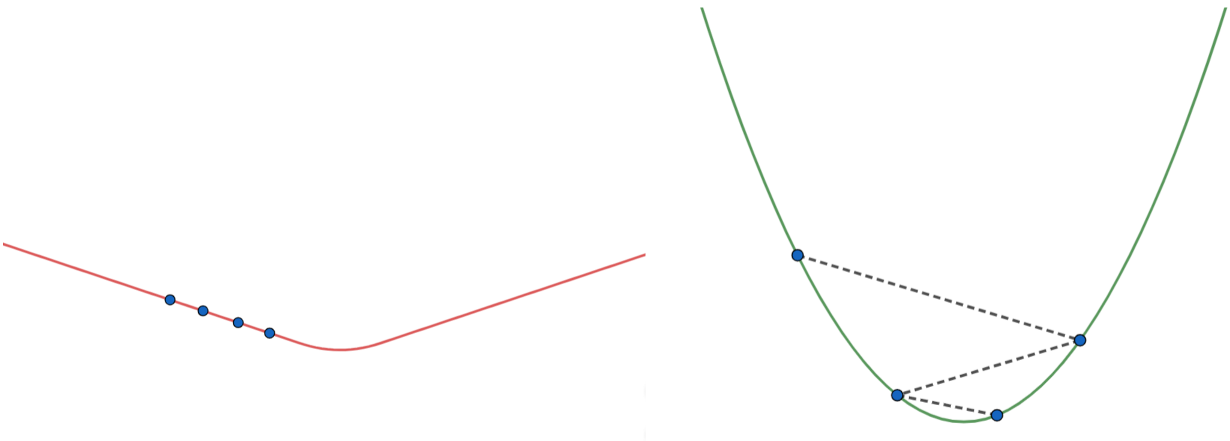
Conjecture 4.1. ([DT14, Conjecture 3.1]). Any sequence of iterates $\{x_i\}$ generated by the gradient method with constant normalized step size $0 \leq h \leq 2$ on a smooth convex function $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$ satisfies

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

We can find a proof for the case $0 \leq h \leq 1$ in [DT14]. The proof for step sizes $1 \leq h \leq 2$ remains open. As explained in [Tay17], the upper bound in this conjecture cannot be improved, since it matches the performance of the gradient method on two specific one-dimensional functions. The first one is piecewise affine-quadratic while the second one is purely quadratic. We will use these functions later to deduce the different worst-case functions wherever the step sizes are not constant. Let define these two functions:

$$\begin{aligned} f_1(x) &= \begin{cases} \frac{LR}{2Nh+1}|x| - \frac{LR^2}{2(2Nh+1)^2} & \text{if } |x| \geq \frac{R}{2Nh+1}, \\ \frac{L}{2}x^2 & \text{else,} \end{cases} \\ f_2(x) &= \frac{L}{2}x^2. \end{aligned} \tag{4.1}$$

We will show later that the function accuracy of the gradient method on f_1 is equal to the first part of the max expression in Conjecture (4.1.) and it is equal to the second part of the max expression on f_2. These two functions can be seen as the worst-case functions for the gradient method with fixed constant step size applied to a L-smooth convex function. In other words, the worst-case behavior of the gradient method with the objective function accuracy criterion is achieved by f_1 or f_2, depending on which of the two is worse. We illustrate the behavior of the gradient method on functions f_1 and f_2 in Figures 4.1a and 4.1b respectively. As we can see, the iterates slowly approach the optimal value of f_1 without overshooting the optimal solution. On the other hand, the iterates of function f_2 overshoot the optimal point at each iteration.



(a) piecewise affine-quadratic function f_1 (b) quadratic function f_2

Figure 4.1: Behavior of the gradient method on functions f_1 and f_2 for $L = R = 1$.

Note that the Conjecture (4.1.) has been verified numerically by solving the formulation (sdp-PEP) in [Tay17].

We can find a new conjecture in [THG17c] that is a generalization of Conjecture (4.1.) for smooth strongly convex functions.

Conjecture 4.2. ([THG17c, Conjecture 2]). Any sequence of iterates $\{x_i\}$ generated by the gradient method with constant normalized step sizes $0 \leq h \leq 2$ on a smooth strongly convex function $f \in \mathcal{F}_{\mu,L}\left(\mathbb{R}^d\right)$ satisfies

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

where $\kappa = \frac{\mu}{L}$ is the inverse of the condition number.

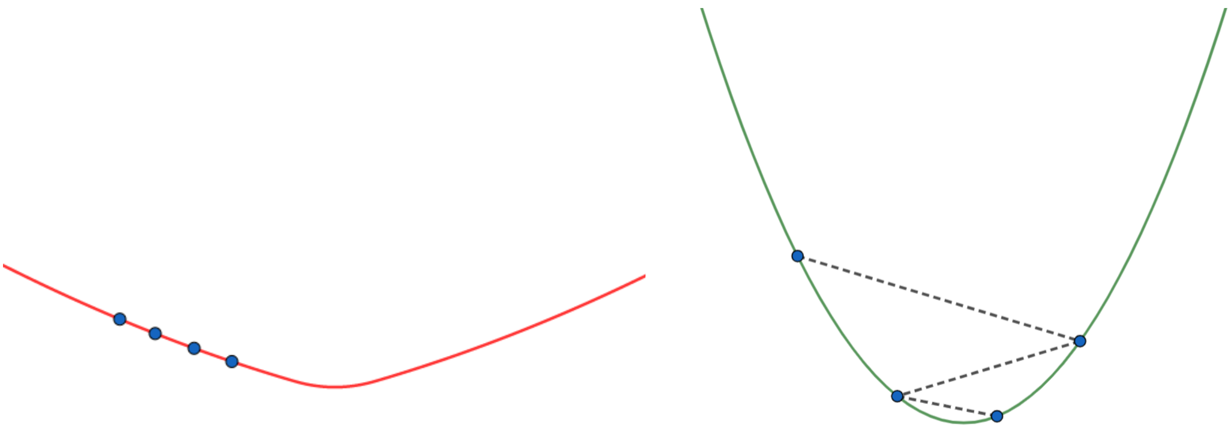
We can find a proof of this conjecture for $N = 1$ in [Tay17, Appendix 4.A]. As for the case of smooth convex functions, we can define two one-dimensional functions that achieve the two parts of the max expression in Conjecture (4.2.):

$$f_{1,\tau}(x) = \begin{cases} \frac{\mu}{2}x^2 + a_\tau|x| + b_\tau & \text{if } |x| \geq \tau \\ \frac{L}{2}x^2 & \text{else} \end{cases}$$
$$f_2(x) = \frac{L}{2}x^2. \tag{4.2}$$

where $a_\tau = (L-\mu)\tau$ and $b_\tau = -\left(\frac{L-\mu}{2}\right)\tau^2$ ensure continuity of $f_{1,\tau}$ and its first derivative. The parameter τ controls the radius of the central quadratic piece. As well explained in [Tay17], the value of τ can be computed analytically by maximizing the objective value of the final iterate $f_{1,\tau}(x_N)$ and we get:

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

The worst-case behavior of the gradient method according to the objective function accuracy is then achieved by $f_{1,\tau}$ or f_2, depending on which of the two is worst. Those behaviors are illustrated on Figure 4.2. The same observations can be made as for the worst-case functions in the smooth convex case. However, the affine pieces of f_1 are replaced by quadratic pieces in $f_{1,\tau}$, as we can see in Figure 4.2a. As for Conjecture (4.1.), these two functions can be seen as the worst-case functions for the gradient method with fixed constant step size applied to smooth strongly convex functions.



(a) Worst-case function $f_{1,\tau}$ (b) Quadratic worst-case function f_2

Figure 4.2: Behavior of the gradient method on functions $f_{1,\tau}$ and f_2 for $L = R = 1$.

Now that we have identified the worst-case functions for fixed constant step sizes, we will use them as a basis to find the worst-case functions for fixed variable step sizes. We will see in the next chapter some new conjectures introduced for the first time in [Dac19]. The idea is then to find a set of functions that achieve the bounds of these conjectures. The two previous conjectures are no longer valid, since we are dealing now with variable step sizes, except for one iteration. This chapter is dedicated to this case. Before that, here is a reminder of the method used.

Gradient Method (GM) with fixed variable step sizes

Input: $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, $x_0 \in \mathbb{R}^d$, $0 \leq h_i \leq 2$
For $i = 1, \ldots, N$

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

4.2 Smooth convex functions

For one step of gradient method, the exact worst-case bound on the objective function accuracy is directly given by Conjecture (4.1.) with $N = 1$. The bound is defined by the maximum of two functions. We will denote these functions $b_{1,1}$ and $b_{1,2}$. More precisely, they are defined as follows:

$$b_{1,1}(h) = \frac{1}{2h+1}$$
$$b_{1,2}(h) = (1-h)^2$$

Therefore, the exact worst-case bound for one iteration of GM is given by

$$b^1(h) = \frac{LR^2}{2}\max(b_{1,1}(h), b_{1,2}(h)) \tag{4.3}$$

The unique step size h will be considered *Small* if $\max(b_{1,1}, b_{1,2}) = b_{1,1}$. Conversely, when h is *Small* we will be in the small step sizes regime. However, if $max(b_{1,1}, b_{1,2}) = b_{1,2}$ the step size will be considered *Big* and we will be in the big step sizes regime. Since we do only one step, the worst-case functions are already known and are defined in equations (4.1) replacing N by one, i.e., they are given by

$$\begin{aligned}\phi_{1,1}(x) &= \begin{cases} \frac{LR}{2h+1}|x| - \frac{LR^2}{2(2h+1)^2} & \text{if } |x| \geq \frac{R}{2h+1}, \\ \frac{L}{2}x^2 & \text{else,}\end{cases} \\ \phi_{1,2}(x) &= \frac{L}{2}x^2\end{aligned} \tag{4.4}$$

Our first objective was to verify those functions numerically. Firstly, we solved the formulation (sdp-PEP) with the PESTO toolbox [THG17a] for hundred and one values of step size h between zero and two defined as follows:

$$h \in \{2i/100\}_{i=0,1,\ldots,100}$$

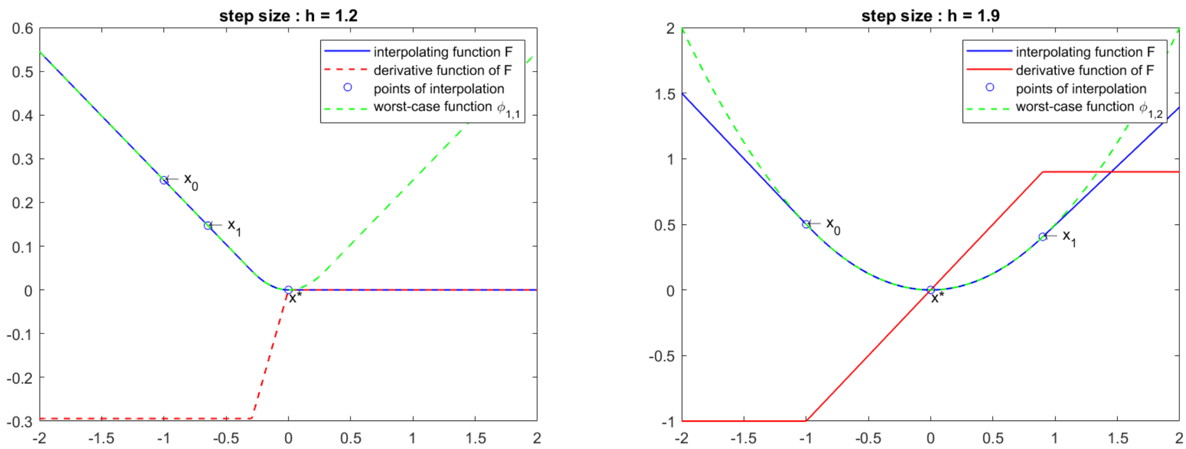
The numerical resolutions of (sdp-PEP) are done using the SDP solver MOSEK [Mos10]. As explained in section (3.3), thanks to the homogeneity of the optimal values of (sdp-PEP) with respect to L and R, we set these parameters to one in our simulations. Each resolution of (sdp-PEP) gives us the worst-case bound associated but also the set of triples $S = \{(x_i, f_i, g_i)\}_{i\in I}$ with $I = \{0, 1, *\}$. The first observation that we made was that all (sdp-PEP) solutions were one-dimensional as functions $\phi_{1,1}$ and $\phi_{1,2}$. Thus, we recovered the set of triples of each (sdp-PEP) problem and we generated our data set with them. Afterwards, we used an interpolation code developed in [KGH21]. It will compute a function $F \in \mathcal{F}_{0,1}(\mathbb{R})$ such that for all $i \in I$ we have $F(x_i) = f_i$ and $g_i \in \partial F(x_i)$. More information about convex interpolation is available in Chapter 2, Section (2.2). The interpolation code works only for one-dimensional inputs. This applies for one iteration of GM, as we discovered.

Since we knew in advance the worst-case bounds to be reached and that they only depend on the step sizes, we directly made groups with the step sizes corresponding to the right worst-case function. For example, the first part of the max expression in equation (4.3), denoted by $b_{1,1}$, is reached for $h \in \{0, 0.5, 0.8, 1, 1.2, 1.4\}$. On the other hand, the bound $b_{1,2}$ is reached for $h \in \{1.6, 1.7, 1.8, 1.9, 2\}$.

Finally, we compared graphically the interpolating function F with the worst-case function associated. Obviously, there are many functions satisfying our conditions of interpolation. Indeed, several functions are 1-smooth convex, respecting the interpolation conditions and having the same final objective value accuracy. Therefore, we only compared the behavior of the two functions on our iterates. Outside them, the two functions may differ. Lastly, for each worst-case function, we also applied one step of GM to prove that the objective function accuracy corresponds to the conjectured worst-case bound.

For example, we can see in Figures 4.3a and 4.3b the interpolating functions as well as the worst-case functions $\phi_{1,1}$ and $\phi_{1,2}$ for respectively $h = 1.2$ and $h = 1.9$ with the parameters L and R set to one. In the sequel, the interpolating function corresponds to the blue segment, its derivative function to the red segment, the worst-case function to the green dotted segment and the iterates to the blue circles. As explained previously, we are only sure that the worst-case function matches the interpolating function perfectly between the iterates. Since there is not a unique worst-case function, the interpolating function outside the iterates can take several shapes, as seen in Figure 4.3a. In Figure 4.3b, we observe that the interpolating function has a linear form outside the iterates. This is possible because there is no strong convexity constraint, as we will see later in this chapter.

The first iterate x_1 in Figure 4.3a stays in the affine part and never comes close to the quadratic part. On the other hand, the iterate x_1 of function $\phi_{1,2}$ overshoots the optimal solution, as seen in Figure 4.3b. This is characteristic of *Big* step sizes. We will discuss this notion in detail later in this chapter. It should also be noted that the starting point x_0 is equal to negative one in our two examples. Since the initial distance to the optimum R is equal to one, the starting point could also be equal to one instead. The worst-case performance according to objective function accuracy is the same if we start with $x_0 = 1$ or $x_0 = -1$. This is confirmed by the fact that the two worst-case functions are symmetric about the y-axis, i.e., they are even. Therefore, if the initial iterate was equal to one, the first iterate x_1 will be located at the opposite of the one in our example.



(a) $\phi_{1,1}(x)$ for $h = 1.2$ (b) $\phi_{1,2}(x)$ for $h = 1.9$

Figure 4.3: The iterates of the GM, the interpolating function and its derivative as well as the corresponding worst-case function for two different step sizes h and with the parameters L and R set to one.

Now that we have compared the two functions graphically, we can easily prove that the worst-case functions $\phi_{1,1}$ and $\phi_{1,2}$ defined in equations (4.4) reach the bounds $b_{1,1}$ and $b_{1,2}$ when we apply one step of GM. We took $x_0 = -R$ as the starting point to remain consistent with our numerical simulations. For the piecewise affine-quadratic function $\phi_{1,1}$ we define the first and unique iterate as follows:

$$\begin{aligned} x_1 &= x_0 - \frac{h}{L}\frac{d\phi_{1,1}}{dx}(x_0) \\ &= -R + \frac{h}{L}\frac{LR}{2h+1} \\ &= \frac{-R(h+1)}{2h+1} \end{aligned}$$

This expression clearly shows that the first iterate remains on the same side of the solution as the starting point. The function accuracy is then given by

$$\begin{aligned} \phi_{1,1}(x_N) - \phi_{1,1}(x_*) &= \phi_{1,1}(x_1) \\ &= \frac{-LR}{2h+1}\frac{-R(h+1)}{2h+1} - \frac{LR^2}{2(2h+1)^2} \\ &= \frac{2LR(R(h+1)) - LR^2}{2(2h+1)^2} \\ &= \frac{LR^2(2h+1}{2(2h+1)^2} \\ &= \frac{LR^2}{2}\frac{1}{2h+1} \end{aligned}$$

It is indeed equal to $\frac{LR^2}{2}b_{1,1}$. For the purely quadratic function, the proof is much simpler. We have $x_1 = -R + \frac{h}{L}LR = -R(1-h)$ and therefore,

$$\phi_{1,2}(x_N) - \phi_{1,2}(x^*) = \frac{LR^2}{2}(1-h)^2$$

which is equal to $\frac{LR^2}{2}b_{1,2}$. We can find in [Tay17, Section 4.3.1] details about the optimal step size for one iteration (optimal in the sense of achieving the lowest worst-case). In this case, we have $h_{opt} = 1.5$. Then for one step, when $h \leq h_{opt}$ we reach the bound $b_{1,1}$ and when $h \geq h_{opt}$ we reach the bound $b_{1,2}$. As already discussed, h is considered *Small* in the first case and *Big* in the second case. Since $h > 1.5$ for function $\phi_{1,2}$, it is clear that the first iterate whose expression is derived just above, overshoots the optimal solution. Finally, we have just proved that the two worst-case functions reach the bounds of Conjecture (4.1.) when the number of iterations is one.

4.3 Smooth strongly convex functions

The exact worst-case bound on the objective accuracy function for one step of GM is given by Conjecture (4.2.) replacing N by one:

$$b^{1,\kappa}(h) = \frac{LR^2}{2}\max\left(\frac{\kappa}{(\kappa-1)+(1-\kappa h)^{-2}}, (1-h)^2\right) \tag{4.5}$$

We will denote the first part of the max expression by $b^{\kappa}_{1,1}$ and the second one by $b^{\kappa}_{1,2}$. The step size h will be considered *Small* if $max(b^{\kappa}_{1,1}, b^{\kappa}_{1,2}) = b^{\kappa}_{1,1}$. Otherwise, the step size will be considered *Big*. As for the smooth convex case, the worst-case functions are already known and are defined in equations (4.2) where we set the number of iterations to one. More precisely,

$$\begin{aligned} \phi^{\mu}_{1,1}(x) &= \begin{cases} \frac{\mu}{2}x^2 + a_\tau|x| + b_\tau & \text{if } |x| \geq \tau \\ \frac{L}{2}x^2 & \text{else} \end{cases} \\ \phi^{\mu}_{1,2}(x) &= \frac{L}{2}x^2. \end{aligned} \tag{4.6}$$

where $a_\tau = (L-\mu)\tau$ and $b_\tau = -\left(\frac{L-\mu}{2}\right)\tau^2$. Only the parameter τ changes and becomes

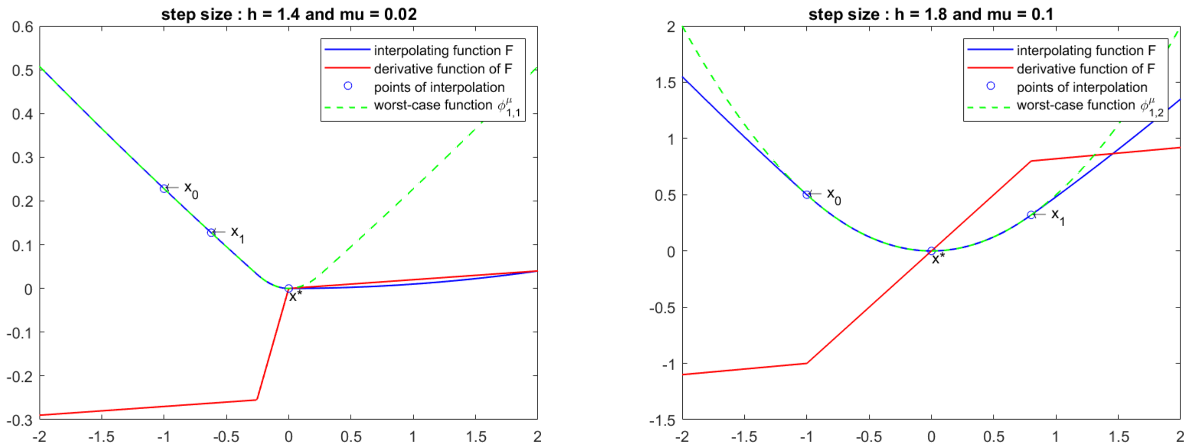
$$\tau = \frac{R\kappa}{(\kappa-1)+(1-\kappa h)^{-2}}$$

To verify those equations numerically, i.e., that the interpolating functions match the worst-case functions, we proceed the same way as for the convex case. Thus, our unique step size h and our strongly convex constant μ used in our simulations are defined as follows:

$$h \in \{2i/100\}_{i=0,1,\dots,100} \text{ and } \mu \in \{i/100\}_{i=0,1,\dots,50}$$

Then, we solved the performance estimation problem (sdp-PEP) for the gradient method for every combination of h and μ. The parameters L and R were set to one. Here too, all our (sdp-PEP) solutions were one-dimensional as functions $\phi^{\mu}_{1,1}$ and $\phi^{\mu}_{1,2}$. Thus, we retrieved the set of triples $\{(x_i, f_i, g_i)\}_{i\in\{0,1,*\}}$ and we generated our data set. The purpose in this case was to find a function $F \in \mathcal{F}_{\mu,L}(\mathbb{R})$ that interpolates the set of triples. In other words, the interpolation code tried to compute a function $F \in \mathcal{F}_{\mu,L}(\mathbb{R})$ such that $F(x_i) = f_i$ and $g_i \in \partial F(x_i)$ for all $i \in \{0, 1, *\}$. We refer to Theorem (2.22.) for more details about strongly convex interpolation. As for the convex case, we compared the interpolating functions and the worst-case functions graphically.

Figures 4.4a and 4.4b show the interpolating function and the corresponding worst-case function for $h = 1.4$ and $h = 1.8$ with strongly convexity constant μ equal to 0.02 and 0.1 respectively. As in the convex case, the interpolating function matches the worst-case function $\phi^{\mu}_{1,1}$ only in the left side of the optimal solution. However, we can observe that the interpolating function in Figure 4.4a takes a small quadratic shape for points of positive abscissa. This can be explained by the strong convexity character. We can do the same observation in Figure 4.4b. The interpolating function has a growing derivative outside the iterates. Because of the strong convexity, there are no more affine parts.



(a) $\phi^{\mu}_{1,1}(x)$ for $(h, \mu) = (1.4, 0.02)$ (b) $\phi^{\mu}_{1,2}(x)$ for $(h, \mu) = (1.8, 0.1)$

Figure 4.4: The iterates of the GM, the interpolating function and its derivative as well as the corresponding worst-case function for two different combinations of (h, μ) and with the parameters L and R set to one.

Since function $\phi^{\mu}_{1,2}$ is equal to function $\phi_{1,2}$ (and therefore the worst-case bound is the same), the proof of showing that we reach the right bound is already made in the previous subsection. For function $\phi^{\mu}_{1,1}$ we can try to do the same reasoning than for $\phi_{1,1}$. We begin by setting L and R to one. The first iterate is then given by

$$\begin{aligned}
x_1 &= x_0 - h\frac{d\phi^{\mu}_{1,1}}{dx}(x_0) \\
&= -1 - h(-\mu - (1-\mu)\tau) \\
&= -1 - h\left(-\mu - \frac{\mu - \mu^2}{(\mu - 1) + (1 - \mu h)^{-2}}\right) \\
&= -1 - h\left(\frac{-\mu(\mu - 1) - \mu(1 - \mu h)^{-2} - (\mu - \mu^2)}{(\mu - 1) + (1 - \mu h)^{-2}}\right)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{-(\mu-1)-(1-\mu h)^{-2}+h\mu(1-h\mu)^{-2}}{(\mu-1)+(1-\mu h)^{-2}} \\
&= \frac{-(\mu-1)-(1-\mu h)^{-2}(1-h\mu)}{(\mu-1)+(1-\mu h)^{-2}} \\
&= \frac{-(\mu-1)-(1-\mu h)^{-1}}{(\mu-1)+(1-\mu h)^{-2}} \\
&= \frac{-\left((\mu-1)+(1-\mu h)^{-1}\right)}{(\mu-1)+(1-\mu h)^{-2}}
\end{aligned}
$$

Therefore, the function accuracy is defined as follows:

$$
\begin{aligned}
\phi^{\mu}_{1,1}(x_N) - \phi^{\mu}_{1,1}(x_*) &= \phi^{\mu}_{1,1}(x_1) \\
&= \frac{\mu}{2}\left(\frac{-((\mu-1)+(1-\mu h)^{-1})}{(\mu-1)+(1-\mu h)^{-2}}\right)^2 + \frac{(\mu-\mu^2)\left((\mu-1)+(1-\mu h)^{-1}\right)}{\left[(\mu-1)+(1-\mu h)^{-2}\right]^2} \\
&\quad - \frac{\mu(\mu-\mu^2)}{2\left[(\mu-1)+(1-\mu h)^{-2}\right]^2} \\
&= \frac{\mu}{2}\frac{(\mu-1)^2+(1-\mu h)^{-2}+2(\mu-1)(1-\mu h)^{-1}}{\left[(\mu-1)+(1-\mu h)^{-2}\right]^2} \\
&\quad + \frac{2(\mu-\mu^2)\left((\mu-1)+(1-\mu h)^{-1}\right)}{2\left[(\mu-1)+(1-\mu h)^{-2}\right]^2} \\
&\quad - \frac{\mu(\mu-\mu^2)}{2\left[(\mu-1)+(1-\mu h)^{-2}\right]^2} \\
&= \frac{\mu(\mu-1)^2+\mu(1-\mu h)^{-2}+2\mu(\mu-1)(1-\mu h)^{-1}}{2\left[(\mu-1)+(1-\mu h)^{-2}\right]^2} \\
&\quad - \frac{2\mu(\mu-1)\left((\mu-1)+(1-\mu h)^{-1}\right)}{2\left[(\mu-1)+(1-\mu h)^{-2}\right]^2} \\
&\quad - \frac{\mu(\mu-\mu^2)}{2\left[(\mu-1)+(1-\mu h)^{-2}\right]^2} \\
&= \frac{-\mu(\mu-1)^2+\mu(1-\mu h)^{-2}+\mu^2(\mu-1)}{2\left[(\mu-1)+(1-\mu h)^{-2}\right]^2} \\
&= \frac{(\mu-1)(-\mu^2+\mu+\mu^2)+\mu(1-\mu h)^{-2}}{2\left[(\mu-1)+(1-\mu h)^{-2}\right]^2} \\
&= \frac{\mu\left((\mu-1)+(1-\mu h)^{-2}\right)}{2\left[(\mu-1)+(1-\mu h)^{-2}\right]^2} \\
&= \frac{1}{2}\frac{\mu}{(\mu-1)+(1-\mu h)^{-2}}
\end{aligned}
$$

This last expression is indeed equal to $\frac{LR^2}{2}b^{\kappa}_{1,1}$ when L and R are equal to one. Now that we have proof for this specific case, we can generalize the proof to any value of these parameters.

The first iterate is expressed as follows:

$$\begin{aligned}
x_1 &= x_0 - \frac{h}{L}\frac{d\phi^{\mu}_{1,1}}{dx}(x_0) \\
&= -R - \frac{h}{L}(-R\mu - (L-\mu)\tau) \\
&= -R - \frac{h}{L}\left(-R\mu - \frac{R(L\mu - \mu^2)/L}{(\kappa-1)+(1-\kappa h)^{-2}}\right) \\
&= -R - \frac{h}{L}\left(\frac{-R\mu(\kappa-1) - R\mu(1-\kappa h)^{-2} - R\mu(1-\kappa)}{(\kappa-1)+(1-\kappa h)^{-2}}\right) \\
&= \frac{-RL(\kappa-1) - RL(1-\kappa h)^{-2} + Rh\mu(1-h\kappa)^{-2}}{L\left((\kappa-1)+(1-\kappa h)^{-2}\right)} \\
&= \frac{-R(\kappa-1) - R(1-\kappa h)^{-2} + Rh\kappa(1-h\kappa)^{-2}}{\left((\kappa-1)+(1-\kappa h)^{-2}\right)} \\
&= \frac{-R(\kappa-1) - R(1-\kappa h)^{-2}(1-h\kappa)}{(\kappa-1)+(1-\kappa h)^{-2}} \\
&= \frac{-R\left((\kappa-1) - (1-\kappa h)^{-1}\right)}{(\kappa-1)+(1-\kappa h)^{-2}}
\end{aligned}$$

Finally, the generalized objective function accuracy is given by

$$\begin{aligned}
\phi^{\mu}_{1,1}(x_N) - \phi^{\mu}_{1,1}(x_*) &= \phi^{\mu}_{1,1}(x_1) \\
&= \frac{\mu}{2}\left(\frac{-R\left((\kappa-1)-(1-\kappa h)^{-1}\right)}{(\kappa-1)+(1-\kappa h)^{-2}}\right)^2 + \frac{R^2\mu(1-\kappa)\left((\kappa-1)+(1-\kappa h)^{-1}\right)}{\left[(\kappa-1)+(1-\kappa h)^{-2}\right]^2} \\
&\quad - \frac{R\frac{\mu^2}{L}(1-\kappa)}{2\left[(\kappa-1)+(1-\kappa h)^{-2}\right]^2} \\
&= \frac{\mu R^2(\kappa-1)^2 + \mu R^2(1-\kappa h)^{-2} + 2\mu R^2(\kappa-1)(1-\kappa h)^{-1}}{2\left[(\kappa-1)+(1-\kappa h)^{-2}\right]^2} \\
&\quad - \frac{2R^2\mu(\kappa-1)\left((\kappa-1)+(1-\kappa h)^{-1}\right)}{2\left[(\kappa-1)+(1-\kappa h)^{-2}\right]^2} \\
&\quad - \frac{R^2\frac{\mu^2}{L}(1-\kappa)}{2\left[(\kappa-1)+(1-\kappa h)^{-2}\right]^2} \\
&= \frac{-R^2\mu(\kappa-1)^2 + R^2\mu(1-\kappa h)^{-2} + R^2\frac{\mu^2}{L}(\kappa-1)}{2\left[(\kappa-1)+(1-\kappa h)^{-2}\right]^2} \\
&= \frac{R^2(\kappa-1)(-\mu\kappa+\mu+\mu\kappa) + R^2\mu(1-\kappa h)^{-2}}{2\left[(\kappa-1)+(1-\kappa h)^{-2}\right]^2} \\
&= \frac{R^2\mu\left((\kappa-1)+(1-\kappa h)^{-2}\right)}{2\left[(\kappa-1)+(1-\kappa h)^{-2}\right]^2}
\end{aligned}$$

$$= \frac{LR^2}{2} \frac{\kappa}{(\kappa - 1) + (1 - \kappa h)^{-2}}$$

We then proved that the worst-case function $\phi_{1,1}^{\mu}$ reaches the bound $\frac{LR^2}{2} b_{1,1}^{\kappa}(h)$ for any value of L and R when applying one step of GM. In the sequel, we will only detail the proofs for such cases. To end this section, we show that the function $\phi_{1,1}^{\mu}$ tends to function $\phi_{1,1}$ as parameter κ tends to zero. We can find in [Dac19, Section 4.2, p. 25] the following for all N and h.

$$\lim_{\kappa \to 0} \frac{\tau}{R} = \lim_{\kappa \to 0} \frac{\kappa}{(\kappa - 1) + (1 - \kappa h)^{-2N}} = \frac{1}{1 + 2Nh} \tag{4.7}$$

Proof. Firstly, we introduce the following variable:

$$q = \frac{1}{(1 - \kappa h)^2}$$

Then, we can write

$$\begin{aligned}
\left(\frac{\kappa}{(\kappa - 1) + (1 - \kappa h)^{-2N}} \right)^{-1} &= \frac{\kappa - 1 + q^N}{\kappa} \\
&= 1 + \frac{1}{\kappa} \left(q^N - 1 \right) \\
&= 1 + \frac{1}{\kappa} \left(\frac{q^N - 1}{q - 1} \right) (q - 1) \\
&= 1 + \frac{1}{\kappa} \left(\frac{1}{q} \sum_{i=1}^{N} q^i \right) (q - 1) \\
&= 1 + \frac{1}{\kappa} \left(\frac{1}{q} \sum_{i=1}^{N} q^i \right) \left(\frac{2\kappa h - \kappa^2 h^2}{(1 - \kappa h)^2} \right) \\
&= 1 + \left(\sum_{i=1}^{N} q^i \right) \left(2h - \kappa h^2 \right)
\end{aligned}$$

When κ tends to zero, q tends to one. Therefore,

$$\lim_{\kappa \to 0} \left(\frac{\kappa}{(\kappa - 1) + (1 - \kappa h)^{-2N}} \right)^{-1} = (1 + 2Nh)$$

Using this relation, our proof becomes straightforward. Indeed,

$$\begin{aligned}
\lim_{\kappa \to 0} \frac{\mu}{2} x^2 + (L - \mu)\tau|x| - \frac{(L - \mu)}{2} \tau^2 &= \lim_{\kappa \to 0} L\tau|x| + \frac{L}{2}\tau^2 \\
&= \lim_{\kappa \to 0} LR \frac{\kappa}{(\kappa - 1) + (1 - \kappa h)^{-2}} |x| - \frac{LR^2}{2} \left(\frac{\kappa}{(\kappa - 1) + (1 - \kappa h)^{-2}} \right)^2 \\
&= \frac{LR}{1 + 2h} |x| - \frac{LR^2}{2(1 + 2h)^2}
\end{aligned}$$

Which is equal to the affine-quadratic piece of worst-case function $\phi_{1,1}$ and valid for $|x| \geq \frac{1}{1+2h}$.

4.4 Conclusion

The aim of this chapter was to derive a set of functions that achieve the worst-case behaviors of one step of GM according to objective function accuracy and that are valid for any value of parameters L and R. For one step of GM on smooth (possibly strongly) convex functions, the two worst-case functions are known and we started by checking them numerically. Then, we applied one iteration of GM on these functions and we computed the final objective value accuracy to confirm that the already known worst-case bound was reached. These functions are summarized in Table 4.1.

	$\mu = 0$	$\mu > 0$ $a = (L-\mu)\tau$ and $b = -\left(\frac{L-\mu}{2}\right)\tau^2$
Small	$\phi_{1,1}(x) = \begin{cases} \frac{LR}{2h_1+1}\lvert x\rvert - \frac{LR^2}{2(2h_1+1)^2}, & \lvert x\rvert \geq \frac{R}{2h_1+1} \\ \frac{L}{2}x^2, & \lvert x\rvert < \frac{1}{2h_1+1} \end{cases}$	$\phi^{\mu}_{1,1}(x) = \begin{cases} \frac{\mu}{2}x^2 + a_\tau\lvert x\rvert + b_\tau, & \lvert x\rvert \geq \tau \\ \frac{L}{2}x^2, & \lvert x\rvert < \tau \end{cases}$ with $\tau = \frac{R\kappa}{(\kappa-1)+(1-\kappa h)^{-2}}$
Big	$\phi_{1,2}(x) = \frac{L}{2}x^2$	$\phi^{\mu}_{1,2}(x) = \frac{L}{2}x^2$

Table 4.1: Worst-case functions depending on the size of the unique step size, for the smooth convex case and the smooth strongly convex case

These functions are used as a basis for deducing those with two and three iterations of GM as we will see in the following chapters.

Chapter 5

Two iterations of gradient method

In this chapter, we will try to derive the worst-case functions for two iterations of the gradient method equipped with the objective function accuracy criterion and with variable normalized step sizes. As previously stated, we will be inspired by the results of previous chapter. Also, we will use conjectures on the behavior of the gradient method developed in the master thesis of A. Daccache [Dac19]. These conjectures are tailored for smooth convex (possibly strongly) unconstrained minimization. The purpose is then to identify the functions that will achieve the tight worst-case bounds provided by the conjectures, when the gradient method is applied to them. Moreover, from now on, we will only detail the proofs for the general case, i.e., for any value of the parameters L and R. As we will see later, we first deduced the worst-case functions for the specific case where $L = R = 1$. Then, by appropriate scaling, we generalized the function for any value of these parameters. Also note that all our numerical examples are with $L = R = 1$. We begin by the case of smooth convex unconstrained minimization.

5.1 Smooth convex functions

We begin by giving the following conjecture developed in [Dac19] for two steps of the gradient method with fixed variable step sizes on smooth convex functions:

Conjecture 5.1. ([Dac19, Conjecture 4.3]). Any sequence of iterates $\{x_1, x_2\}$ generated by two steps of the gradient method with variable normalized step sizes $(h_1, h_2) \in [0, 2] \times [0, 2]$ on a smooth convex function $f \in \mathcal{F}_{0,L}\left(\mathbb{R}^d\right)$ satisfies:

$$f(x_2) - f_* \leq \frac{LR^2}{2} \max\left(\frac{1}{1 + 2h_1 + 2h_2}, (1 - h_1)^2 (1 - h_2)^2, \right. \\ \left. \frac{1}{(1 + h_1)^2} (1 - h_2)^2, \frac{1}{1 + 2h_2} (1 - h_1)^2\right) \tag{5.1}$$

This conjecture is not yet proven.[1] However, we can find in [Dac19] a proof for the first part of the max expression for step sizes such that $(h_1, h_2) \in [0, 1]^2$. This is given in the following theorem.

[1]A beginning of proof can be found in [Dac19, Section 5.2].

Theorem 5.1. [Dac19, Theorem 5.5]. Consider a sequence $\{x_1, x_2\}$ generated by two step sizes of the classical gradient method on a smooth convex function $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$ then the exact worst-case bound on the objective function accuracy for all $(h_1, h_2) \in [0, 1]^2$ is given by

$$f(x_2) - f(x_*) \leq \frac{LR^2}{2} \frac{1}{1 + 2h_1 + 2h_2}$$

For two fixed variable step sizes of GM, we consider that there are four types of combination between the step sizes (h_1, h_2) which are

(S,S),(B,B),(S,B) and *(B,S)*

Where S corresponds to a *Small* step size and B to a *Big* step size. We admit then four different regimes for the worst-case performance. Each regime corresponds to a type of combination. The worst-case bound according to the objective function accuracy is then given by

$$b^2(h_1, h_2) = \frac{LR^2}{2} \max(b_{2,1}(\mathbf{h}), b_{2,2}(\mathbf{h}), b_{2,3}(\mathbf{h}), b_{2,4}(\mathbf{h}))$$

Where $\mathbf{h} = (h_1, h_2)$. The $\{b_{2,i}\}_{i\in\{1,2,2,4\}}$ are equal to the terms in the max expression of Conjecture (5.1.). These terms correspond to the worst-case bounds of the four regimes. Encouraged by the results for one step of GM, we can make the following assumption on the number of worst-case functions.

Assumption 5.1. Each of the four functions $\{b_{2,i}\}_{i\in\{1,2,2,4\}}$ that compose the worst-case bound of Conjecture (5.1.) is respectively reached by applying two steps of GM on respectively four different worst-case functions.

Our way of proceeding will be as follows. As for one iteration, we begin by solving the performance estimation problem (sdp-PEP) for the gradient method for several values of step sizes and according to the function accuracy criterion. In our simulations, we set L and R to one, because of the homogeneity of the optimal values (see Section 3.3). The step sizes h_1 and h_2 are defined as follows:

$$(h_1, h_2) \in \{2i/100\}_{i=0,1,\ldots,100}$$

The first observed result was that all problems were one-dimensional. Thus, we can make the following assumption on the dimension of the worst-case functions.

Assumption 5.2. The worst-case performance according to the objective function accuracy of two steps of GM with fixed variable step sizes such that $(h_1, h_2) \in [0, 2]$, is achieved by a one-dimensional function.

Since the iterates are one-dimensional, we can use the interpolation code developed in [KGH21] again. The purpose is to find for each combination of step sizes, a function $F \in \mathcal{F}_{0,1}(\mathbb{R})$ that interpolates the set of triples $\{x_i, f_i, g_i\}_{i\in\{0,1,2,*\}}$ given by the PESTO toolbox [THG17a]. We refer to Corollary (2.23.) for the interpolation conditions that the set of triples must meet.

Once we found the interpolating functions, the first step was to try to deduce the worst-case functions graphically. As we will see, the technique involves looking at the derivatives of the interpolating functions. The interpolating functions are piecewise functions with affine and purely quadratic parts. The purpose is then to look at the slope of the derivative functions and where they change shape. Once we thought we had the right worst-case function, we applied two steps of GM with h_1 and h_2 to verify if the objective function accuracy reaches the corresponding bound. Then, we generalized the function to any value of parameters L and R and applied the GM again. Since we knew in advance the worst-case bounds to be reached, we also grouped the combinations of step sizes with the right bound. We can now begin with the first region.

5.1.1 Region 1 (S,S)

The first region corresponds to the regime *(S,S)* and the bound associated is the first part of the max expression in Conjecture (5.1.), denoted by b_1. More precisely,

$$b_{2,1}(h_1, h_2) = \frac{1}{1 + 2h_1 + 2h_2}$$

Intuitively we could imagine that the worst-case function will have the same behavior as the piecewise affine-quadratic function $\phi_{1,1}$ defined in equations (4.4). For two constant step sizes, the worst-case function is given by replacing h by $h + h = 2h$ in equations (4.4). So we suppose that for two variable step sizes, we can replace h by $h_1 + h_2$. We get the following piecewise worst-case function:

$$\phi_{2,1}(x) = \begin{cases} \frac{RL}{2h_1+2h_2+1}|x| - \frac{R^2L}{2(2h_1+2h_2+1)^2}, & |x| \geq \frac{R}{2h_1+2h_2+1} \\ \frac{L}{2}x^2, & |x| < \frac{R}{2h_1+2h_2+1} \end{cases} \tag{5.2}$$

We see in Figure 5.1 the iterates of the GM, the interpolating function and the worst-case function $\phi_{2,1}(x)$ for $(h_1, h_2) = (1.2, 1.4)$ and $L = R = 1$. Here too, the interpolating function matches with the worst-case function between the iterates but takes a different way outside them, i.e., on the right of its axis of symmetry. As function $\phi_{1,1}$, the iterates stay in the first affine part and the global shape is the same. Only the slope of the two linear parts changes. We will see later that for three step sizes belonging to region *(S,S,S)*, the worst-case function has the same behavior. It should also be noted that the iterates never approach the quadratic piece.

We can demonstrate that our guessed worst-case function achieves the worst-case bound $\frac{LR^2}{2}b_{2,1}$ when we apply two steps of GM. Since the initial distance to the solution is R, the starting point x_0 can take the values R or $-R$ for example. In all our simulations, the starting point is $x_0 = -1$. To remain consistent with this, we set $x_0 = -R$ in our proofs. It will play a role for the next regions, as we will see later in this chapter. Thus, for the first iterate we have:

$$x_1 = x_0 - \frac{h_1}{L}\frac{d\phi_{2,1}}{dx}(x_0) = -R + \frac{h_1}{L}\frac{LR}{2(h_1 + h_2) + 1} = \frac{-R(2h_2 + h_1 + 1)}{2(h_1 + h_2) + 1}$$

The second iterate is given by

$$\begin{aligned} x_N = x_2 &= x_1 - \frac{h_2}{L}\frac{d\phi_{2,1}}{dx}(x_1) \\ &= \frac{-R(2h_2 + h_1 + 1)}{2(h_1 + h_2) + 1} + \frac{h_2}{L}\frac{LR}{2(h_1 + h_2) + 1} \end{aligned}$$

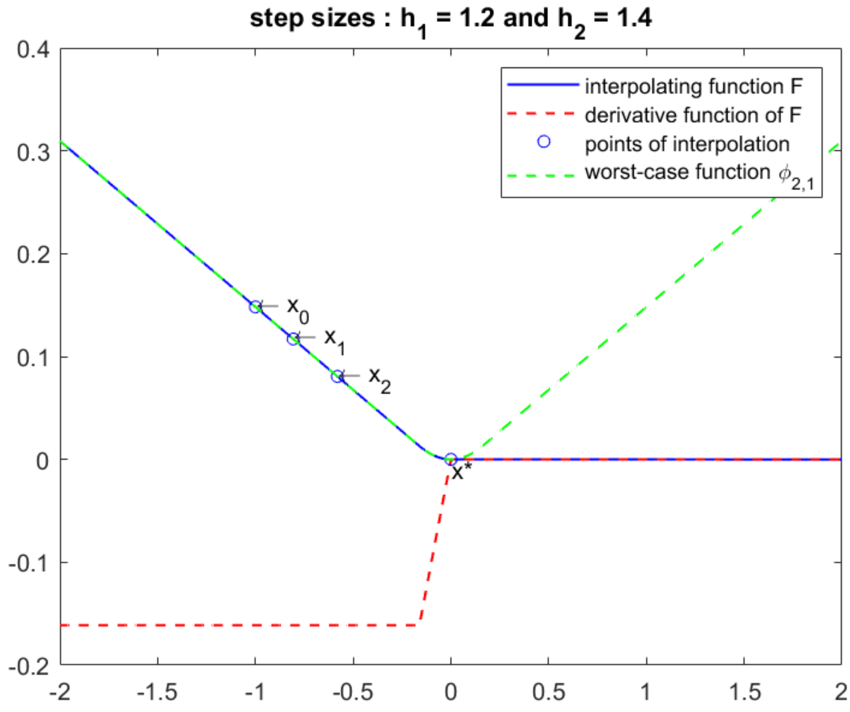


Figure 5.1: The iterates of the GM, the interpolating function and its derivative as well as the worst-case function $\phi_{2,1}$ for $(h_1, h_2) = (1.2, 1.4)$ with the parameters L and R set to one.

$$= \frac{-R(h_2 + h_1 + 1)}{2(h_1 + h_2) + 1}$$

It is clear that the iterates stay on the same side of the solution. Finally, we compute the objective function accuracy that is given by

$$\begin{aligned}
\phi_{2,1}(x_N) - \phi_{2,1}(x_*) &= \phi_{2,1}(x_N) \\
&= \frac{LR^2(h_2 + h_1 + 1)}{(2(h_1 + h_2) + 1)^2} - \frac{LR^2}{2(2(h_1 + h_2) + 1)^2} \\
&= \frac{LR^2(2h_2 + 2h_1 + 1)}{2(2h_1 + 2h_2 + 1)^2} \\
&= \frac{LR^2}{2} \frac{1}{2(h_1 + h_2) + 1}
\end{aligned}$$

This last expression is indeed equal to $\frac{LR^2}{2} b_{2,1}$. We can now move on to the second region.

5.1.2 Region 2 (B,B)

The region *(B,B)* corresponds to the second part of the max expression in Conjecture (5.1.), defined as follows:

$$b_{2,2}(h_1, h_2) = (1 - h_1)^2 (1 - h_2)^2$$

For two constant step sizes, the worst-case bound is given by Conjecture (4.1.) with N equal to two, i.e., equal to $(1 - h)^2 (1 - h)^2$. For two variables step sizes we only replace the two occurrences of h by h_1 and h_2 respectively. Since the worst-case function for constant step sizes defined in equations (4.1) does not depend on the step size h, we can imagine that it is the same case for fixed variable step sizes. The worst-case function is then equal to $\phi_{1,2}$:

$$\phi_{2,2}(x) = \frac{Lx^2}{2} \tag{5.3}$$

We will see later that for three different step sizes in the region *(B,B,B)*, the worst-case function is the same. We will also generalize to N iterations with step sizes that belong to the region with only *Big* step sizes. Figure 5.2 shows the interpolating function and the worst-case function $\phi_{2,2}$ for the case where $(h_1, h_2) = (1.8, 1.6)$ and $L = R = 1$. Here too, the interpolating function takes a linear shape outside the iterates.

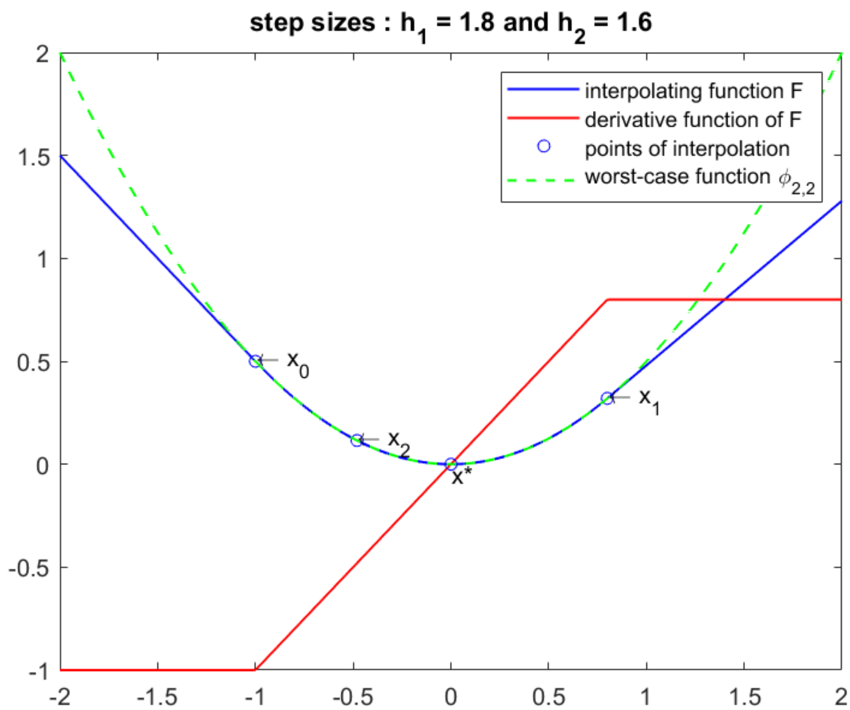


Figure 5.2: The iterates of the GM, the interpolating function and its derivative as well as the worst-case function $\phi_{2,2}$ for $(h_1, h_2) = (1.8, 1.6)$ with the parameters L and R set to one.

We can easily show that our worst-case function reaches the worst-case bound $\frac{LR^2}{2} b_{2,2}$ when we apply two steps of GM. We have for the first iterate:

$$\begin{aligned} x_1 &= x_0 - \frac{h_1}{L}\frac{d\phi_{2,2}}{dx}(x_0) \\ &= -R + \frac{h_1}{L} LR \\ &= -R(1 - h_1) \end{aligned}$$

Iterate x_2 is given by

$$x_2 = x_1 - \frac{h_2}{L}\frac{d\phi_{2,2}}{dx}(x_1)$$

$$= -R(1-h_1) + \frac{h_2}{L} LR(1-h_1)$$
$$= -R(1-h_1)(1-h_2)$$

Therefore,

$$\phi_{2,2}(x_N) = \frac{LR^2}{2}(1-h_1)^2(1-h_2)^2$$

5.1.3 Region 3 (B,S)

The third region corresponds to the third bound of the max expression in Conjecture (5.1.). More precisely,

$$b_{2,3} = \frac{(h_1-1)^2}{(2h_2+1)}$$

The first step size is considered to be *Big* while the second one is considered to be *Small*. Figure 5.3 shows the starting point x_0, the two iterates x_1 and x_2 as well as the optimal solution x_*. As we can see, the first iterate x_1 computed with the *Big* step size h_1 goes to the other side of the optimal solution (i.e., overshoots the solution). Then, the second iterate x_2 computed with the *Small* step size h_2 stays in the affine part and does not oscillate around the solution.

We can suppose that the first piece of the worst-case function is purely quadratic and equal to $\phi_{2,2}$ until we reach the affine part. In other words, the function is quadratic until its derivative becomes constant. The worst-case function is then characterized by a single change in the form of the derivative. Since the derivative of the quadratic part is equal to $\phi'_{2,2}(x) = x$ when $L = 1$, at the intersection of the two pieces we have that $\frac{d\phi^-_{2,3}}{dx}(m) = m$ where m denotes the slope of the linear part. The minus sign indicates that the starting point is negative. We found out that the slope is equal to $\frac{h_1-1}{2h_2+1}$. We guessed this slope by setting $h_1 = 2$ and $h_2 = 1$. Therefore, by appropriate scaling, we defined the following worst-case function valid for any value of L and R:

$$\phi^-_{2,3}(x) = \begin{cases} \frac{LR(h_1-1)}{2h_2+1}x - \frac{LR^2}{2}\left(\frac{h_1-1}{2h_2+1}\right)^2, & x \geq \frac{R(h_1-1)}{2h_2+1} \\ \frac{L}{2}x^2, & x < \frac{R(h_1-1)}{2h_2+1} \end{cases} \tag{5.4}$$

Figure 5.3 shows the interpolating function and the worst-case function for $(h_1, h_2) = (1.8, 0.8)$ and $L = R = 1$. The interpolating function is linear before the starting point and then matches perfectly the worst-case function. The difficulty here was that the change of the derivative function did not occur at one of the iterates, as is the case in the next region.

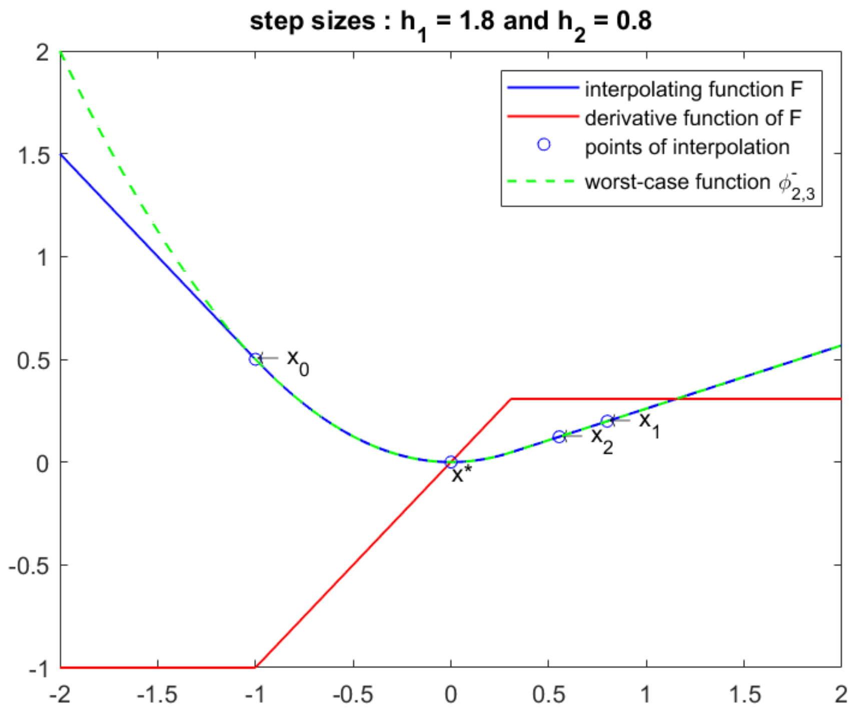


Figure 5.3: The iterates of the GM, the interpolating function and its derivative as well as the worst-case function $\phi^-_{2,3}$ for $(h_1, h_2) = (1.8, 0.8)$ with the parameters L and R set to one.

Now that we have deduced graphically the worst-case function, we can demonstrate that the objective function accuracy reaches the bound $\frac{LR^2}{2}b_{2,3}$ when we apply two iterations of GM. But first it is useful to ask what happens if the starting point is instead equal to 1 in our example. The two previous worst-case functions from first and second regions have the property to be even (i.e, $f(x) = f(-x)$). In this case, if the starting point is equal to one, the worst-case function remains the same. For the third region, the worst-case function is no longer even. Figure **??** shows the interpolating function and the worst-case function associated when the starting point x_0 is equal to one. We can see that the worst-case function is obtained by symmetry of the graph of $\phi^-_{2,3}$ around the y axis, i.e., exchange positive and negative abscissas $(f(x) \to f(-x))$. We will denote this new function $\phi^+_{2,3}$ to indicate that the starting point is positive.

Let's return to our proof and define the first iterate as follows:

$$x_1 = x_0 - h_1 \frac{d\phi^-_{2,3}}{dx}(x_0) = -R - \frac{h_1}{L}(-LR) = R(h_1 - 1)$$

The second iterate is defined as:

$$\begin{aligned} x_2 &= x_1 - \frac{h_2}{L}\frac{d\phi^-_{2,3}}{dx}(x_1) \\ &= R(h_1 - 1) - \frac{h_2}{L}\frac{LR(h_1 - 1)}{2h_2 + 1} \\ &= \frac{R(h_1 - 1)(h_2 + 1)}{2h_2 + 1} \end{aligned}$$

Therefore, the objective function accuracy is given by

$$\begin{aligned}\phi^-_{2,3}(x_N) &= \frac{(h_1-1)^2 LR^2(h_2+1)}{2h_2+1} - \frac{LR^2}{2}\left(\frac{h_1-1}{2h_2+1}\right)^2 \\ &= \frac{(h_1-1)^2 LR^2(2h_2+2-1)}{2(2h_2+1)^2} \\ &= \frac{LR^2}{2}\frac{(h_1-1)^2}{(2h_2+1)}\end{aligned}$$

This last expression is in fact equal to $\frac{LR^2}{2}b_{2,3}$. Finally, the worst-case function for any value of L and R when the starting point is positive, is given by

$$\phi^+_{2,3}(x) = \begin{cases} \frac{LR(1-h_1)}{2h_2+1}x - \frac{LR^2}{2}\left(\frac{h_1-1}{2h_2+1}\right)^2, & x \le \frac{R(1-h_1)}{2h_2+1} \\ \frac{L}{2}x^2, & x > \frac{R(1-h_1)}{2h_2+1} \end{cases}. \tag{5.5}$$

5.1.4 Region 4 (S,B)

The last region corresponds to the fourth part of the max expression in Conjecture (5.1.). As a reminder, it is given by

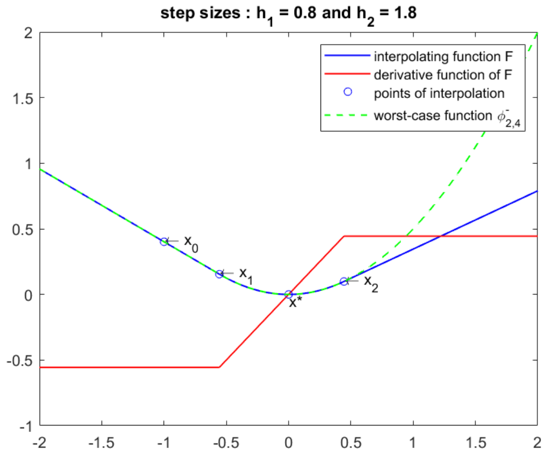
$$b_{2,4} = \frac{(1-h_2)^2}{(1+h_1)^2}$$

The first step size is considered to be *Small* while the second step size is considered to be *Big*. As we can see in Figure 5.4a, the first iterate x_1 stays in the affine part of the interpolating function. Then, we have an overshoot of the optimal solution with the second iterate x_2. As we saw for the third region, a *Big* step size corresponds to an overshoot of the optimum, while a *Small* step size corresponds to a simply descent on the affine part to approach the solution.

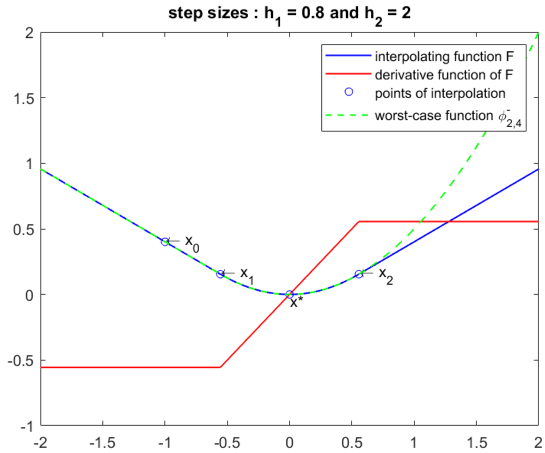
As for the second region, we can imagine that the worst-case function at the right of the optimal solution will be purely quadratic and will not follow the second affine part of the interpolating function. This quadratic piece will be the same function as in the second region, i.e., $\phi_{2,2}(x) = \frac{Lx^2}{2}$. Therefore, looking in Figure 5.4a at the interpolating function and its derivative, we realize that we only need to find the slope of the first linear part to describe the worst-case function.

Firstly, we discovered with our numerical simulations that the slope does not depend on h_2, as we can see in Figures 5.4a and 5.4b. The function is continuous and the derivative of the quadratic part is $\phi'_{2,2}(x) = x$ when $L = 1$. Also, graphically we can see that the change of shape seems to take place at the first iterate x_1. So we know that at the intersection of the two pieces (i.e, at the change of derivative expression) we have that $\frac{d\phi^-_{2,4}}{dx}(m) = x_1$, where m denotes the slope of the affine piece. As for the previous region, the minus sign indicates the starting point x_0 is negative and equal to -1 in our example. We found out that the slope is equal to $\frac{-1}{1+h_1}$. We guessed this slope by testing with $h_1 = 0.5$ and $h_1 = 1$. By appropriate scaling, we defined the worst-case function as:

$$\phi^-_{2,4}(x) = \begin{cases} \frac{-LR}{1+h_1}x - \frac{LR^2}{2(1+h_1)^2}, & x \le \frac{-R}{1+h_1} \\ \frac{L}{2}x^2, & x > \frac{-R}{1+h_1} \end{cases} \tag{5.6}$$



(a) $(h_1, h_2) = (0.8, 1.8)$



(b) $(h_1, h_2) = (0.8, 2)$

Figure 5.4: The iterates of the GM, the interpolating function and its derivative as well as the worst-case function $\phi_{2,4}^{-}$ for two different combinations of step sizes with the parameters L and R set to one.

We are now able to prove that the worst-case bound $\frac{LR^2}{2} b_{2,4}$ is achieved by function $\phi_{2,4}^{-}$ when we apply two iterations of GM. The first iterate is defined as:

$$x_1 = x_0 - \frac{h_1}{L}\frac{d\phi_{2,4}^{-}}{dx}(x_0) = -R + \frac{Rh_1}{1+h_1} = \frac{-R}{1+h_1}$$

And the second one as:

$$\begin{aligned} x_2 &= x_1 - \frac{h_2}{L}\frac{d\phi_{2,4}^{-}}{dx}(x_1) \\ &= \frac{-R}{1+h_1} + h_2\frac{R}{1+h_1} \\ &= \frac{-R(1-h_2)}{1+h_1} \end{aligned}$$

Finally, the objective function accuracy is given by

$$\phi_{2,4}^{-}(x_N) = \frac{LR^2}{2}\frac{(1-h_2)^2}{(1+h_1)^2}$$

This last expression is in fact equal to $\frac{LR^2}{2} b_{2,4}$. As with the previous region, the question arises as to what happens if the starting point is no longer equal to minus one but rather equal to one. We clearly observe in Figure 5.4 that the worst-case function is not symmetric around the y axis. Indeed, if $x_0 = 1$, the first iterate will stay on the affine part on the right side of the optimum while the second iterate will be on the other side, i.e., it will overshoot the solution. This new worst-case function is obtained by orthogonal symmetry around the y axis of function $\phi_{2,4}^{-}$. Figure 5.5 shows an example of the interpolating and worst-case function in this particular case. The worst-case function valid for any value of R and L with a positive starting point is given by

$$\phi^+_{2,4}(x) = \begin{cases} \frac{RL}{1+h_1}x - \frac{RL^2}{2(1+h_1)^2}, & x \geq \frac{R}{1+h_1} \\ \frac{L}{2}x^2, & x < \frac{R}{1+h_1} \end{cases} \tag{5.7}$$

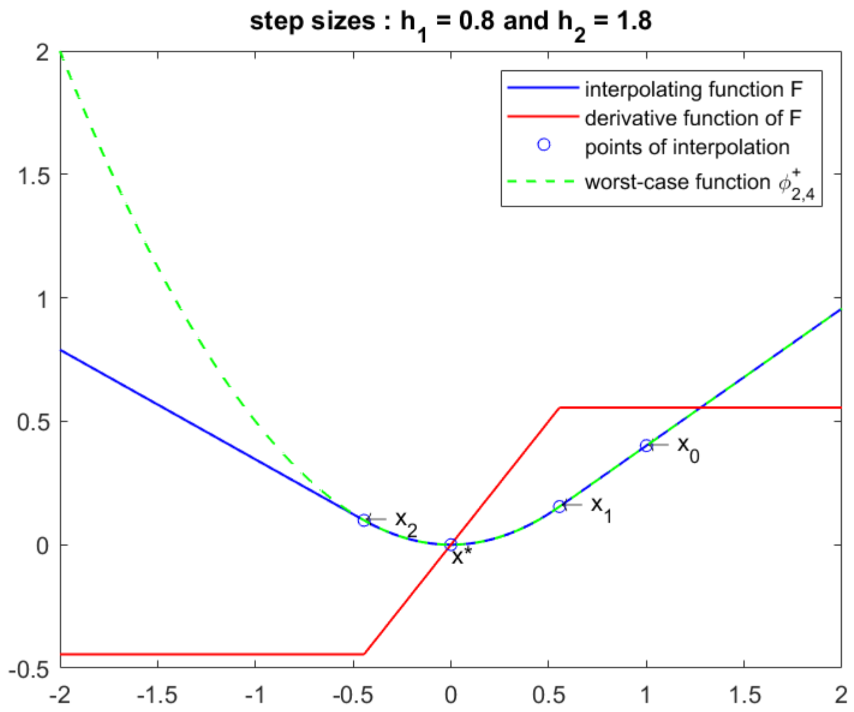


Figure 5.5: The iterates of the GM, the interpolating function and his derivative as well as the worst-case function $\phi^+_{2,4}$ for $(h_1, h_2) = (0.8, 1.8)$ and the starting point $x_0 = 1$.

5.2 Smooth strongly convex functions

For two steps of the gradient method on smooth strongly convex functions with fixed variable step sizes, we begin with the following conjecture developed in [Dac19].

Conjecture 5.2. Any sequence of iterates $\{x_1, x_2\}$ generated by two steps of gradient method with variable normalized step sizes $(h_1, h_2) \in [0, 2] \times [0, 2]$ on a smooth strongly convex function

$f \in \mathcal{F}_{\mu,L}\left(\mathbb{R}^d\right)$ satisfies:

$$f(x_2) - f_* \leq \frac{LR^2}{2} \max\Bigg(\frac{\kappa}{(\kappa-1) + (1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}}, \\ (1-h_1)^2(1-h_2)^2, \\ \left(\frac{\kappa}{(\kappa-1)+(1-\kappa h_1)^{-1}}\right)^2 (1-h_2)^2, \\ \frac{\kappa}{(\kappa-1)+(1-\kappa h_2)^{-2}}(1-h_1)^2 \Bigg) \tag{5.8}$$

As explained in [Dac19], there are four different regimes in terms of h_1 and h_2 for each value of κ. Each regime corresponds to a type of combination between the step sizes. Those types of combination are the same as for the convex case:

$$(S,S),(B,B),(B,S) \text{ and } (S,B)$$

Each regime corresponds to a bound of the max expression in Conjecture (5.2.). We will denote the worst-case bound of the above Conjecture as follows:

$$b^{2,\kappa}(h_1,h_2) = \frac{LR^2}{2}\max(b^{\kappa}_{2,1}, b^{\kappa}_{2,2}, b^{\kappa}_{2,3}, b^{\kappa}_{2,4})$$

The four functions $\{b^{\kappa}_{2,i}\}_{i\in\{1,2,3,4\}}$ denote respectively the worst-case bound of each regime. It should also be noted that the functions $b^{\kappa}_{2,i}$ tend to the functions $b_{2,i}$ when κ tends to zero. We can find a proof in [Dac19]. From this, we can suppose that Assumption (5.1.) is also valid for the smooth strongly convex case and we will denote those four worst-case functions by $\phi^{\mu}_{2,i}$ with $i \in \{1,2,3,4\}$. Encouraged by the results for one step of GM, we can make the following assumption on the convergence of the worst-case functions.

Assumption 5.3. Consider two variable step sizes of GM. We have the following link between the worst-case functions based on the objective function accuracy in the smooth convex and smooth strongly cases for all $i \in J = \{1,2,3,4\}$:

$$\lim_{\kappa\to 0} \phi^{\mu}_{2,i} = \phi_{2,i}$$

We will prove this statement for each region. Finally, we proceed the same way as for the smooth convex case. Firstly, we solved the (sdp-PEP) for each combination of step sizes and constant μ. We set also the parameters L and R to one. For our simulations, we took fifty one values of step sizes (h_1,h_2) and twenty one values of μ. They are defined as follows:

$$(h_1,h_2) \in \{4i/100\}^2_{i=0,1,\ldots,50} \text{ and } \mu \in \{i/100\}_{i=0,1\ldots,20}$$

As for the convex case, we observe that all problems were one-dimensional. Therefore, Assumption (5.2.) is valid also for smooth strongly convex functions. For the interpolation part, the interpolating function should belong to the class of L-smooth μ-strongly convex functions. More precisely, the function $F \in \mathcal{F}_{\mu,L}(\mathbb{R})$ that interpolates the sets of triples is such that $F(x_i) = f_i$

and $g_i \in \partial F(x_i)$. We refer to Theorem (2.22.) for more details about the conditions to be met.

Since we knew in advance the four bounds to be reached, we made groups of combinations corresponding to the bounds. Once we thought we had found the worst-case function, we applied two steps of GM to see if we had reached the right worst-case bound. We can begin by looking at the first region.

5.2.1 Region 1 (S,S)

The first region corresponds to the first part of the max expression in Conjecture (5.2.). More precisely,

$$b_{2,1}^{\kappa} = \frac{1}{2}\frac{\kappa}{(\kappa-1)+(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}}$$

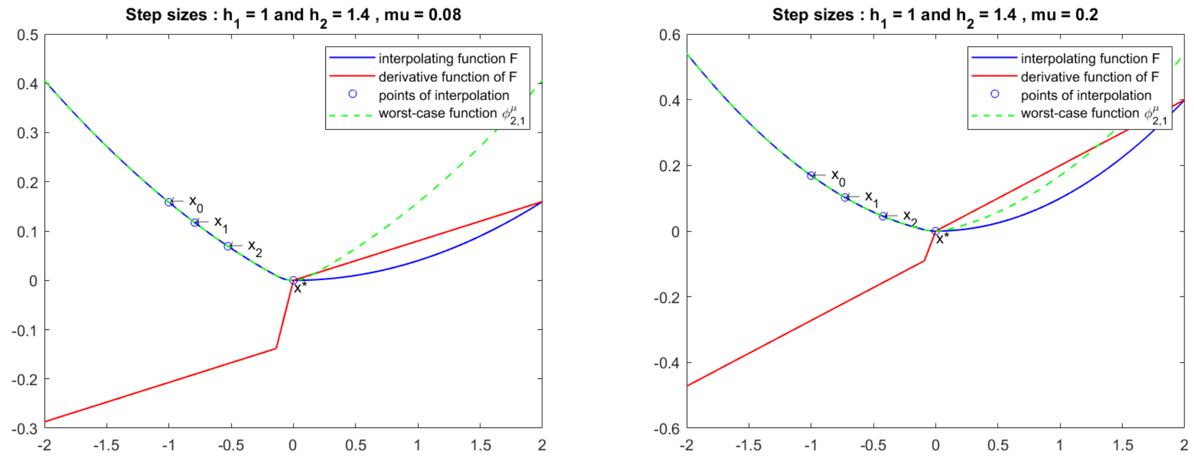
As for the smooth convex case in the region *(S,S)*, the iterates do not oscillate around the optimal solution, as we can see in Figure 5.6. This is the characteristic of *Small* step sizes. However, the interpolating functions do not have more affine parts, since they belong to the class of smooth strongly convex functions. The worst-case functions are already known for constant step sizes. They have been defined in equations (4.2). For two constant step sizes considered to be *Small*, the worst-case function is given by replacing $(1-\kappa h)^{-2}$ by $(1-\kappa h)^{-2}(1-\kappa h)^{-2}$ in the parameter τ in the definition of $\phi_{1,1}^{\mu}$. Then, for two variable step sizes, we can imagine that the worst-case function is given by replacing the two occurrences of h by h_1 and h_2 respectively. As a result, we define the worst-case function for the first region as follows

$$\phi_{2,1}^{\mu}(x) = \begin{cases} \frac{\mu}{2}x^2 + a_\tau|x| + b_\tau & \text{if } |x| \geq \tau \\ \frac{L}{2}x^2 & \text{else} \end{cases} \tag{5.9}$$

where $a_\tau = (L-\mu)\tau$, $b_\tau = -\left(\frac{L-\mu}{2}\right)\tau^2$ and

$$\tau = \frac{R\kappa}{(\kappa-1)+(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}}$$

Figure 5.6 shows the interpolating function and worst-case function for two different values of constant μ with same step sizes. We can observe in Figure 5.6b that for a bigger value of strong convexity constant, the growth of interpolating function is bounded by a better quadratic lower bound. Finally, the same observation can be made about the iterates as for the region (S,S) in the convex case. They stay in the first piece of the worst-case function and never come close to the purely quadratic part $\frac{x^2}{2}$.



(a) $\mu = 0.08$ (b) $\mu = 0.2$

Figure 5.6: The iterates of the GM, the interpolating function and its derivative as well as the worst-case function $\phi^{\mu}_{2,1}$ for two different values of constant μ with the parameters L and R set to one.

We can try to prove that our guessed worst-case function reaches the worst-case bound $\frac{LR^2}{2}b^{\kappa}_{2,1}$ when we apply two steps of GM. We have for the first iterate:

$$\begin{aligned}
x_1 &= x_0 - \frac{h_1}{L}\frac{d\phi^{\mu}_{2,1}}{dx}(x_0) \\
&= -R - \frac{h_1}{L}(-R\mu - (L-\mu)\tau) \\
&= -R - \frac{h_1}{L}\left(-R\mu - \frac{R\mu(1-\kappa)}{(\kappa-1)+(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}}\right) \\
&= -R - \frac{h_1}{L}\left(\frac{-R\mu(\kappa-1) - R\mu(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2} - R\mu(1-\kappa)}{(\kappa-1)+(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}}\right) \\
&= \frac{-R(\kappa-1) - R(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2} + h_1\kappa(1-h_1\kappa)^{-2}(1-\kappa h_2)^{-2}}{(\kappa-1)+(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}} \\
&= \frac{-R(\kappa-1) - R(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}(1-h_1\kappa)}{(\kappa-1)+(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}} \\
&= \frac{-R(\kappa-1) - R(1-\kappa h_1)^{-1}(1-\kappa h_2)^{-2}}{(\kappa-1)+(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}} \\
&= \frac{-R\left((\kappa-1) + (1-\kappa h_1)^{-1}(1-\kappa h_2)^{-2}\right)}{(\kappa-1)+(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}}
\end{aligned}$$

The second iterate is simply given by

$$\begin{aligned}
x_2 &= x_1 - \frac{h_2}{L}\frac{d\phi^{\mu}_{2,1}}{dx}(x_1) \\
&= \frac{-R\left((\kappa-1) + (1-\kappa h_1)^{-1}(1-\kappa h_2)^{-1}\right)}{(\kappa-1)+(1-\mu h_1)^{-2}(1-\kappa h_2)^{-2}}
\end{aligned}$$

Therefore, the objective function accuracy is defined as follows:

$$\begin{aligned}
\phi^{\mu}_{2,1}(x_N) - \phi^{\mu}_{2,1}(x_*) &= \phi^{\mu}_{2,1}(x_2) \\
&= \frac{\mu}{2}\left(\frac{-R\left((\kappa-1)+(1-\kappa h_1)^{-1}(1-\kappa h_2)^{-1}\right)}{(\kappa-1)+(1-\mu h_1)^{-2}(1-\kappa h_2)^{-2}}\right)^2 \\
&\quad + \frac{R^2\mu(1-\kappa)\left((\kappa-1)+(1-\kappa h_1)^{-1}(1-\kappa h_2)^{-1}\right)}{\left[(\kappa-1)+(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}\right]^2} \\
&\quad - \frac{R^2\frac{\mu^2}{L}(1-\kappa)}{2\left[(\kappa-1)+(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}\right]^2} \\
&= \frac{R^2\mu(\kappa-1)^2 + R^2\mu(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2} + 2R^2\mu(\kappa-1)(1-\kappa h_1)^{-1}(1-\kappa h_2)^{-1}}{2\left[(\kappa-1)+(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}\right]^2} \\
&\quad - \frac{2R^2\mu(\kappa-1)\left((\kappa-1)+(1-\kappa h_1)^{-1}(1-\kappa h_2)^{-1}\right)}{2\left[(\kappa-1)+(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}\right]^2} \\
&\quad - \frac{R^2\frac{\mu^2}{L}(1-\kappa)}{2\left[(\kappa-1)+(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}\right]^2} \\
&= \frac{R^2(\kappa-1)(-\mu\kappa+\mu+\mu\kappa) + R^2\mu(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}}{2\left[(\kappa-1)+(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}\right]^2} \\
&= \frac{R^2\mu\left((\kappa-1)+(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}\right)}{2\left[(\kappa-1)+(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}\right]^2} \\
&= \frac{LR^2}{2}\frac{\kappa}{(\kappa-1)+(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}}
\end{aligned}$$

This last expression is in fact equal to the bound $\frac{LR^2}{2}b^{\kappa}_{2,1}$. Before going to the next region, we can prove that the worst-case function $\phi^{\mu}_{2,1}$ tends to function $\phi_{2,1}$ when κ tends to zero. Firstly we have that

$$\begin{aligned}
\lim_{\kappa\to 0}\frac{\tau}{R} &= \lim_{\kappa\to 0}\frac{\kappa}{(\kappa-1)+(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}} \\
&= \frac{1}{1+2(h_1+h_2)}
\end{aligned}$$

The proof is given by setting $q = \frac{1}{(1-\kappa h_1)^2(1-\kappa h_2)^2}$ in the proof of expression (4.7). Therefore, proving our statement becomes straightforward. Indeed,

$$\begin{aligned}
\lim_{\kappa\to 0}\frac{\mu}{2}x^2 + (L-\mu)\tau|x| - \frac{L-\mu}{2}\tau^2 &= L\tau|x| - L\tau^2 \\
&= \frac{LR}{2h_1+2h_1+1}|x| - \frac{LR^2}{2(2h_1+2h_2+1)^2}
\end{aligned}$$

Which is valid when $|x| \geq \frac{1}{1+2(h_1+h_2)}$ and corresponds to the affine piece of function $\phi_{1,1}$.

5.2.2 Region 2 (B,B)

The second region corresponds to the second part of the max expression in Conjecture (5.2.), denoted by $b^{\kappa}_{2,2}$. The bound is the same as in the convex case. Our worst-case function is then also equal to function $\phi_{2,2}$. We can directly generalize the function for any value of L and R as follows:

$$\phi^{\mu}_{2,2}(x) = \frac{Lx^2}{2} \tag{5.10}$$

Figure 5.7 shows the interpolating function and the worst-case function when $(h_1, h_2) = (1.4, 2)$ and $\mu = 0.1$. Unlike the convex case, the interpolating function does not have a linear shape outside the iterates but rather a quadratic form. This is because a strongly convex function can not have an affine piece since its second derivative can not be zero.

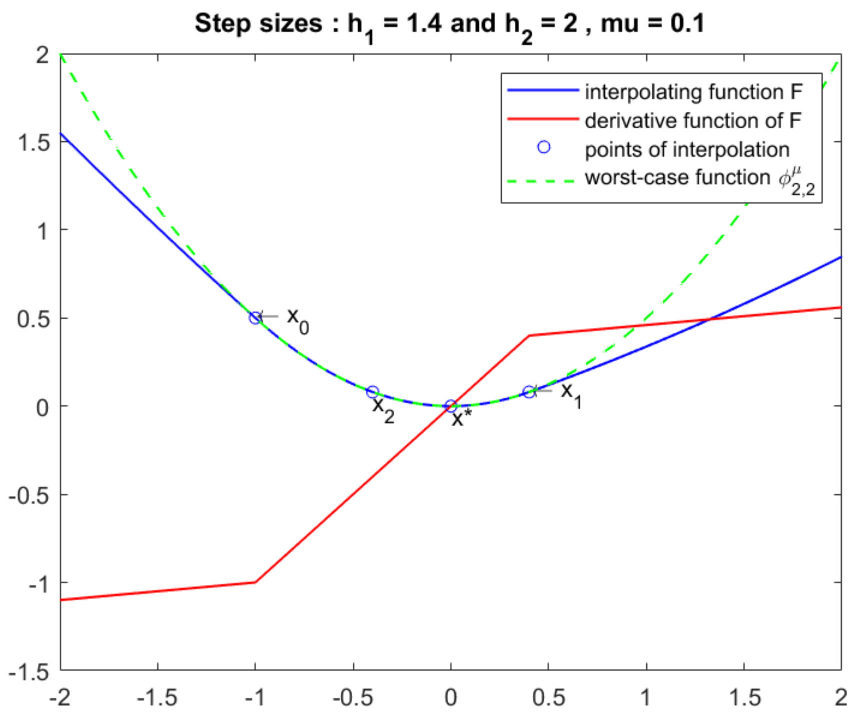


Figure 5.7: The iterates of the GM, the interpolating function and its derivative as well as the worst-case function $\phi^{\mu}_{2,2}$ for $(h_1, h_2, \mu) = (1.4, 2, 0.1)$ with the parameters L and R set to one.

The proof of showing that the bound $\frac{LR^2}{2} b^{\kappa}_{2,2}$ is achieved by function $\phi^{\mu}_{2,2}$ is the same as for worst-case function $\phi_{2,2}$.

5.2.3 Region 3 (B,S)

The third region corresponds to the third part of the max expression in Conjecture (5.2.). More precisely,

$$b^{\kappa}_{2,3}(h_1, h_2) = \frac{\kappa(h_1 - 1)^2}{(\kappa - 1) + (1 - \kappa h_2)^{-2}}$$

This region consists of a first step of *Big* size and a second step of *Small* size. As for the third region in the convex case, the iterates x_1 and x_2 visible in Figure 5.8, are located at the right side of the optimal solution. Indeed, the first step size h_1 results in a oscillation around the optimum. Then, the second step size h_2 brings us down to the solution. As for function $\phi^{-}_{2,3}$, we can suppose that the first piece is equal to $\frac{Lx^2}{2}$ until a change in the form of the derivative near second iterate x_2. This is illustrated in Figure 5.8. The term $(1-\kappa h_1)^{-2}$ seems to correspond to a *Small* step size. Therefore, we try to use the parameter τ from the first region without the $(1-\kappa h_1)^{-2}$ term and by multiplying it by $(h_1 - 1)$, as we did for the third region in the convex case. We then define the worst-case function as follows:

$$\phi^{\mu,-}_{2,3}(x) = \begin{cases} \frac{\mu}{2}x^2 + a_\tau x + b_\tau & \text{if } x \geq \tau \\ \frac{L}{2}x^2 & \text{else} \end{cases} \tag{5.11}$$

where $a_\tau = (L-\mu)\tau$, $b_\tau = -\left(\frac{L-\mu}{2}\right)\tau^2$ and

$$\tau = \frac{R\kappa(h_1 - 1)}{(\kappa - 1) + (1-\kappa h_2)^{-2}}$$

Figure 5.8 shows the interpolating function and the worst-case function for $h_1 = 1.6$, $h_2 = 1$ and $\mu = 0.14$.

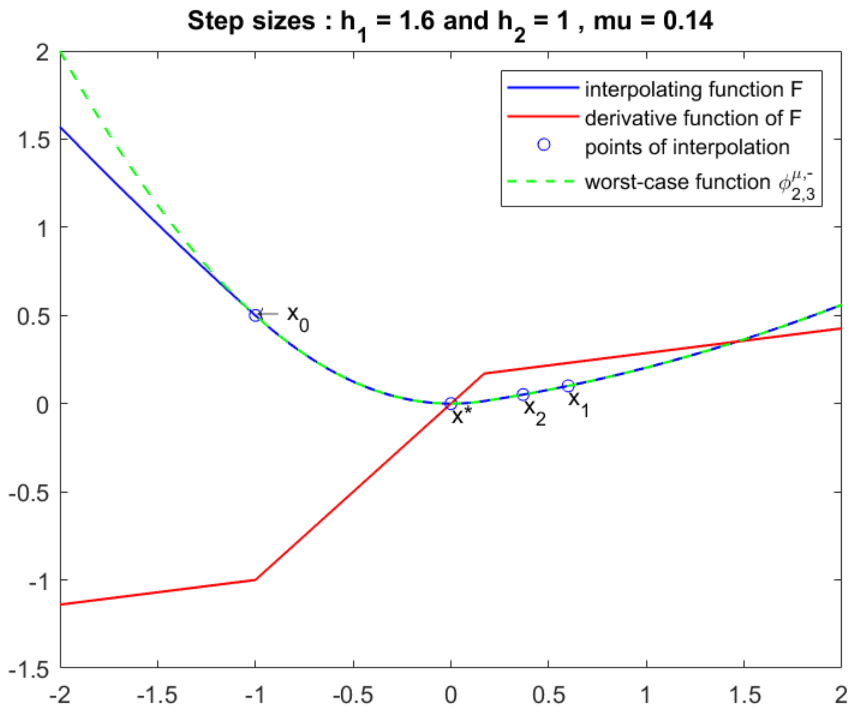


Figure 5.8: The iterates of the GM, the interpolating function and its derivative as well as the worst-case function $\phi^{\mu,-}_{2,3}$ for $(h_1, h_2, \mu) = (1.6, 1, 0.14)$ with the parameters L and R set to one.

As for the first region, we can prove that the worst-case function reaches the right bound. The minus sign in the notation $\phi^{\mu,-}_{2,3}$ denotes that the starting point is negative. It is equal to minus one in our example. As for the third and fourth regions in the convex case, the worst-case function is not symmetric around the solution. Consequently, the worst-case function $\phi_{2,3}^{\mu,-}$ depends on the sign of the initial iterate and therefore it is valid only for negative values of the latter. Since there is not a unique worst-case function, we will restrict ourselves to the function defined above. However, it should be noted that the function valid for positive starting points is obtained by orthogonal symmetry around the y-axis of function $\phi_{2,3}^{\mu,-}$. In our specific case, the first iterate is given by

$$x_1 = x_0 - \frac{h_1}{L}\frac{d\phi_{2,3}^{\mu,-}}{dx}(x_0) = R(-1+h_1)$$

Then, the second iterate x_2 is defined as follows:

$$\begin{aligned}
x_2 &= x_1 - h_2\frac{d\phi_{2,3}^{\mu,-}}{dx}(x_1)\\
&= x_1(1-h_2\kappa) - \frac{Rh_2\kappa(1-\kappa)(h_1-1)}{(\kappa-1)+(1-\kappa h_2)^{-2}}\\
&= \frac{R(\kappa-1)(h_1-1)(1-\kappa h_2) + R(h_1-1)(1-\kappa h_2)^{-1} + Rh_2\kappa(\kappa-1)(h_1-1)}{(\kappa-1)+(1-\kappa h_2)^{-2}}\\
&= \frac{R(h_1-1)\left((\kappa-1)+(1-\kappa h_2)^{-1}\right)}{(\kappa-1)+(1-\kappa h_2)^{-2}}
\end{aligned}$$

Finally, the objective function accuracy in function of parameters L and R is given by

$$\begin{aligned}
\phi_{2,3}^{\mu,-}(x_N) - \phi_{2,3}^{\mu,-}(x^*) = \phi_{2,3}^{\mu,-}(x_2)
&= \frac{\mu}{2}\left(\frac{R(h_1-1)\left((\kappa-1)+(1-\kappa h_2)^{-1}\right)}{(\kappa-1)+(1-\kappa h_2)^{-2}}\right)^2\\
&\quad - \frac{2R^2\mu(\kappa-1)(h_1-1)^2\left((\kappa-1)+(1-\kappa h_2)^{-1}\right)}{2\left[(\kappa-1)+(1-\kappa h_2)^{-2}\right]^2}\\
&\quad - \frac{R^2\frac{\mu^2}{L}(1-\kappa)(h_1-1)^2}{2\left[(\kappa-1)+(1-\kappa h_2)^{-2}\right]^2}\\
&= \frac{R^2\mu(h_1-1)^2\left((\kappa-1)^2+(1-\kappa h_2)^{-2}\right)}{2\left[(\kappa-1)+(1-\kappa h_2)^{-2}\right]^2} - \frac{2R^2\mu(\kappa-1)^2(h_1-1)^2}{2\left[(\kappa-1)+(1-\kappa h_2)^{-2}\right]^2}\\
&\quad - \frac{R^2\frac{\mu^2}{L}(1-\kappa)(h_1-1)^2}{2\left[(\kappa-1)+(1-\kappa h_2)^{-2}\right]^2}\\
&= \frac{R^2\mu(h_1-1)^2\left(-(\kappa-1)^2+(1-\kappa h_2)^{-2}\right)}{2\left[(\kappa-1)+(1-\kappa h_2)^{-2}\right]^2} + \frac{R^2\frac{\mu^2}{L}(\kappa-1)(h_1-1)^2}{2\left[(\kappa-1)+(1-\kappa h_2)^{-2}\right]^2}\\
&= \frac{R^2\mu(h_1-1)^2(\kappa-1)(-(\kappa-1)+\kappa) + R^2\mu(h_1-1)^2(1-\kappa h_2)^{-2}}{2\left[(\kappa-1)+(1-\kappa h_2)^{-2}\right]^2}\\
&= \frac{R^2\mu(h_1-1)^2\left(\kappa-1)+(1-\kappa h_2)^{-2}\right)}{2\left[(\kappa-1)+(1-\kappa h_2)^{-2}\right]^2}\\
&= \frac{LR^2}{2}\frac{\kappa(h_1-1)^2}{(\kappa-1)+(1-\kappa h_2)^{-2}}
\end{aligned}$$

Which is in fact equal to the bound $\frac{LR^2}{2}b^{\kappa}_{2,3}$. To finish with this region, we show that the worst-case function $\phi^{\mu,-}_{3,2}$ tends to function $\phi^{-}_{2,3}$ when κ tends to zero. We already know that

$$\lim_{\kappa\to 0}\frac{\kappa}{(\kappa-1)+(1-\kappa h_2)^{-2}}=\frac{1}{1+2h_2}$$

Therefore

$$\begin{aligned}\lim_{\kappa\to 0}\frac{\mu}{2}x^2+(L-\mu)\tau x-\frac{(L-\mu)}{2}\tau^2 &= \lim_{\kappa\to 0} L\tau x-\frac{L}{2}\tau^2\\ &= \lim_{\kappa\to 0}\frac{LR\kappa(h_1-1)}{(\kappa-1)+(1-\kappa h_2)^{-2}}x-\frac{LR^2}{2}\left(\frac{\kappa(h_1-1)}{(\kappa-1)+(1-\kappa h_2)^{-2}}\right)^2\\ &= \frac{LR(h_1-1)}{2h_2+1}x-\frac{LR^2(h_1-1)^2}{2(2h_2+1)^2}\end{aligned}$$

This function piece is valid for $x\geq\tau$. Since $\lim_{\kappa\to 0}\tau=\frac{R(h_1-1)}{1+2h_2}$, we tend to function $\phi^{-}_{2,3}$. We can now move on to the last region.

5.2.4 Region 4 (S,B)

The fourth region corresponds to the fourth part of the max expression in Conjecture (5.2.), defined as follows:

$$b^{\mu}_{2,4}=\left(\frac{\kappa}{(\kappa-1)+(1-\kappa h)^{-1}}\right)^2(1-h_2)^2$$

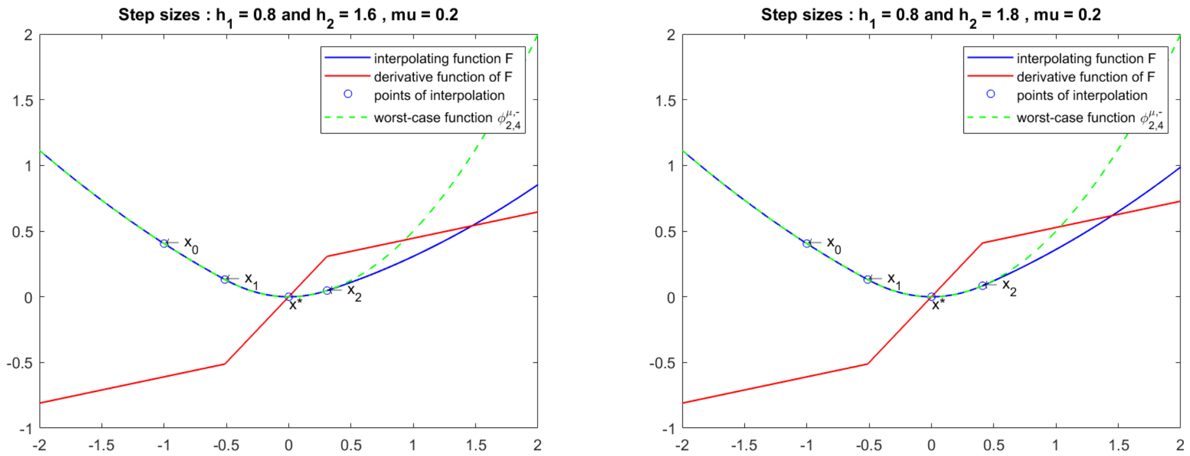
We see two examples of interpolating functions and iterates when $(h_1,h_2,\mu)=(0.8,1.6,0.2)$ in Figure 5.9a and $(h_1,h_2,\mu)=(0.8,1.8,0.2)$ in Figure 5.9b. We can observe that the first step size h_1 does not make the iterate x_1 oscillate around the optimum unlike the second step size h_2. We can interpret the interpolating function as a piecewise function with a particular piece, until a change in the form of the derivative. After which, we fall back on the well-known quadratic function. Also, we found out that the first piece does not depend on h_2. Indeed, the derivative functions in Figure 5.9a and Figure 5.9b have the same slope on the first piece for different step size h_2. Since the first part of the worst-case function does not depend on h_2, we tried taking the parameter τ from the first region, removing the $(1-\kappa h_2)^2$ term and replacing $(1-\kappa h_1)^{-2}$ by $(1-\kappa h_1)^{-1}$. We got the following worst-case function:

$$\phi^{\mu,-}_{2,4}(x)=\begin{cases}\frac{\mu}{2}x^2-a_\tau x+b_\tau & \text{if } x\leq-\tau\\ \frac{L}{2}x^2 & \text{else}\end{cases}\tag{5.12}$$

where $a_\tau=(L-\mu)\tau$, $b_\tau=-\left(\frac{L-\mu}{2}\right)\tau^2$ and

$$\tau=\frac{R\kappa}{(\kappa-1)+(1-\kappa h_1)^{-1}}$$

The worst-case functions associated with our two examples are also plotted in Figures 5.9a and 5.9b. We can now prove that function $\phi^{\mu,-}_{2,4}$ reaches the bound $\frac{LR^2}{2}b^{\mu}_{2,4}$. As in the previous



(a) $h_2 = 1.6$ (b) $h_2 = 1.8$

Figure 5.9: The iterates of the GM, the interpolating function and its derivative as well as the worst-case function $\phi_{2,4}^{\mu,-}$ for two different values of h_2 with the parameters L and R set to one.

section, the minus sign indicates that the starting point must be negative. Therefore, we set $x_0 = -R$. Then, the first iterate is given by

$$\begin{aligned}
x_1 &= x_0 - \frac{h_1}{L}\frac{d\phi_{2,4}^{\mu,-}}{dx}(x_0) \\
&= -R - \frac{h_1}{L}(-R\mu - (L-\mu)\tau) \\
&= -R - \frac{h_1}{L}\left(-R\mu - \frac{R\mu(1-\kappa)}{(\kappa-1)+(1-\kappa h)^{-1}}\right) \\
&= -R - \frac{h}{L}\left(\frac{-R\mu(\kappa-1) - R\mu(1-\kappa h)^{-1} - R\mu(1-\kappa)}{(\kappa-1)+(1-\kappa h)^{-1}}\right) \\
&= -R - h\left(\frac{-R\kappa(1-\kappa h)^{-1}}{(\kappa-1)+(1-\kappa h)^{-1}}\right) \\
&= \frac{-R(\kappa-1) - R(1-\kappa h)^{-1} + Rh\kappa(1-h\kappa)^{-1}}{(\kappa-1)+(1-\kappa h)^{-1}} \\
&= \frac{-R(\kappa-1) - R(1-\kappa h)^{-1}(1-h\kappa)}{(\kappa-1)+(1-\kappa h)^{-1}} \\
&= \frac{-R\kappa}{(\kappa-1)+(1-\kappa h)^{-1}} \\
&= -\tau
\end{aligned}$$

The second iterate x_2 is defined as:

$$\begin{aligned}
x_2 &= x_1 - \frac{h_2}{L}\frac{d\phi_{2,4}^{\mu,-}}{dx}(x_1) \\
&= -\tau - \frac{h_2}{L}(-L\tau)
\end{aligned}$$

$$= -\tau(1-h_2)$$

Finally, the objective function accuracy is given by

$$\begin{aligned}
\phi_{2,4}^{\mu,-}(x_N) - \phi_{2,4}^{\mu,-}(x^*) &= \phi_{2,4}^{\mu,-}(x_2) \\
&= \frac{L\tau^2}{2}(1-h_2)^2 \\
&= \frac{LR^2}{2}\left(\frac{\kappa}{(\kappa-1)+(1-\kappa h)^{-1}}\right)^2 (1-h_2)^2
\end{aligned}$$

This last expression is in fact equal to $\frac{LR^2}{2}b_{2,4}^{\kappa}$. We finish this section by showing that this last worst-case function tends to function $\phi_{2,4}^{-}$ when κ tends to zero. Firstly, we have that

$$\begin{aligned}
\lim_{\kappa\to 0}\frac{\tau}{R} &= \lim_{\kappa\to 0}\frac{\kappa}{(\kappa-1)+(1-\kappa h_1)^{-1}} \\
&= \frac{1}{1+h_1}
\end{aligned}$$

We can find the proof by setting $q = \frac{1}{(1-\kappa h_1)}$ and replacing $-2N$ by $-N$ in expression (4.7). Therefore we have:

$$\begin{aligned}
\lim_{\kappa\to 0}\frac{\mu}{2}x^2 - (L-\mu)\tau x - \frac{(L-\mu)}{2}\tau^2 &= \lim_{\kappa\to 0} -L\tau x - \frac{L}{2}\tau^2 \\
&= \lim_{\kappa\to 0}\frac{-LR\kappa}{(\kappa-1)+(1-\kappa h_1)^{-1}}x - \frac{LR^2}{2}\left(\frac{\kappa}{(\kappa-1)+(1-\kappa h_1)^{-1}}\right)^2 \\
&= \frac{-LR}{h_1+1}x - \frac{LR^2}{2(h_1+1)^2}
\end{aligned}$$

This function piece is valid for $x \leq -\tau$. Since τ tends to $\frac{1}{1+h_1}$ when κ tends to zero, we indeed fall back on function $\phi_{2,4}^{-}$.

5.3 Conclusion

The aim of this chapter was to derive a set of functions that achieve the worst-case behaviors of two steps of GM according to objective function accuracy and that are valid for any value of parameters L and R. For two steps of GM, our results rely on the use of Conjecture (5.1.) for smooth convex functions and Conjecture (5.2.) for smooth strongly convex functions. These conjectures are expressed as the maximum of four functions, depending on the step sizes. The purpose was then to derive a worst-case function for each of these four bounds. Each function is such that the final objective value accuracy of two steps of GM applied to it, is equal to one of the four bounds. Also, we found that the worst-case functions were one-dimensional. The eight functions are summarized in Table 5.1. We have also shown that the worst-case functions in the strongly convex case converge to their equivalent in the convex case.

	$\mu = 0$	$\mu > 0$ $a = (L-\mu)\tau$ and $b = -\left(\frac{L-\mu}{2}\right)\tau^2$
(S, S)	$\phi_{2,1}(x) = \begin{cases} \frac{LR}{2(h_1+h_2)+1}\lvert x\rvert - \frac{LR^2}{2(2(h_1+h_2)+1)^2}, & \lvert x\rvert \geq \frac{R}{2(h_1+h_2)+1} \\ \frac{L}{2}x^2, & \lvert x\rvert < \frac{1}{2(h_1+h_2)+1} \end{cases}$	$\phi^{\mu}_{2,1}(x) = \begin{cases} \frac{\mu}{2}x^2 + a_\tau\lvert x\rvert + b_\tau, & \lvert x\rvert \geq \tau \\ \frac{L}{2}x^2, & \lvert x\rvert < \tau \end{cases}$ with $\tau = \frac{R\kappa}{(\kappa-1)+(1-\kappa h_1)^{-2}(1-\kappa h_2)^{-2}}$
(B, B)	$\phi^{R,L}_{1,2}(x) = \frac{L}{2}x^2$	$\phi^{\mu}_{1,2}(x) = \frac{L}{2}x^2$
(B, S)	$\phi^{-}_{2,3}(x) = \begin{cases} \frac{LR(h_1-1)}{2h_2+1}x - \frac{LR^2}{2}\left(\frac{h_1-1}{2h_2+1}\right)^2, & x \geq \frac{R(h_1-1)}{2h_2+1} \\ \frac{L}{2}x^2, & x < \frac{R(h_1-1)}{2h_2+1} \end{cases}$	$\phi^{\mu,-}_{2,3}(x) = \begin{cases} \frac{\mu}{2}x^2 + a_\tau\lvert x\rvert + b_\tau, & x \geq \tau \\ \frac{L}{2}x^2, & x < \tau \end{cases}$ with $\tau = \frac{R\kappa(h_1-1)}{(\kappa-1)+(1-\kappa h_2)^{-2}}$
(S, B)	$\phi^{-}_{2,4}(x) = \begin{cases} \frac{-LR}{1+h_1}x - \frac{LR^2}{2(1+h_1)^2}, & x \leq \frac{-R}{1+h_1} \\ \frac{L}{2}x^2, & x > \frac{-R}{1+h_1} \end{cases}$	$\phi^{\mu,-}_{2,4}(x) = \begin{cases} \frac{\mu}{2}x^2 + a_\tau\lvert x\rvert + b_\tau, & x \leq -\tau \\ \frac{L}{2}x^2, & x > -\tau \end{cases}$ with $\tau = \frac{R\kappa}{(\kappa-1)+(1-\kappa h_1)^{-1}}$

Table 5.1: Worst-case functions for two steps of GM for the smooth convex case and the smooth strongly convex case.

Chapter 6

Three iterations of gradient method

In this chapter, we will try to identify the worst-case functions for three iterations of the gradient method equipped with the objective function accuracy criterion and with variable normalized step sizes. As for the previous chapter, we will only detail the proofs for the general case, i.e., for any value of the parameters L and R. Also note that all our numerical examples are with $L = R = 1$.

6.1 Smooth convex functions

We can find some results on worst-case performances for three steps of the gradient method according to the function accuracy in the master thesis of A. Daccache [Dac19]. Firstly, it is considered that the number of functions that compose a possible conjecture is equal to 2^N with N the number of iterations. Thus for three steps, we shall consider eight functions. We denote by $b_{3,i}$ the i^{th} function. Each of these functions is maximum for some combinations of step sizes. The exact worst-case bound is given by the maximum of those eight functions. We consider then eight possible types of combination between the three step sizes:

$$(S, S, S)\,(S, B, B)\,(S, S, B)\,(S, B, S)\,(B, S, S)\,(B, S, B)\,(B, B, B) \text{ and } (B, B, S)$$

Where S denotes a *Small* step size and B a *Big* one. Unfortunately, we can only find the worst-case bounds associated with the first seven regimes. We start by identifying the worst-case functions corresponding to these bounds. We proceeded the same way as for two steps. Firstly we solved the performance estimation problems with the PESTO toolbox [THG17a] with $N = 3$, $R = 1$ and $L = 1$. The three step sizes used were defined as follows:

$$(h_1, h_2, h_3) \in \{4i/100\}^3_{i=0,1,\ldots,50}$$

The first interesting result that emerged was that not all problems were one-dimensional. For example, the numerical resolution of sdp-PEP with $(h_1, h_2, h_3) = (2, 1.8, 0.3)$ was two-dimensional. We will see later in this section that problems that are not in 1-D belong to the eighth region, i.e., combinations of type (B, B, S). Knowing the bounds to be reached, we made groups with the combinations of step sizes associated with each bound. We then tried to deduce the different worst-case functions graphically with the interpolation code developed in [KGH21]. Once we had deduced the function, we applied three steps of the gradient method to verify if we landed on the right bound. We began with the worst-case bounds where the first step size h_1 is considered to be *Small*. We find in [Dac19, Section 4.3] these bounds:

$$\begin{aligned} b_{3,1}(h_1,h_2,h_3) &= \frac{1}{1+2h_1+2h_2+2h_3} \\ b_{3,2}(h_1,h_2,h_3) &= \left(\frac{1}{(1+h_1)^2}\right)(1-h_2)^2(1-h_3)^2 \\ b_{3,3}(h_1,h_2,h_3) &= \left(\frac{1}{(1+h_1+h_2)^2}\right)(1-h_3)^2 \\ b_{3,4}(h_1,h_2,h_3) &= \left(\frac{1}{(1+h_1)^2}\right)(1-h_2)^2\left(\frac{1}{1+2h_3}\right) \end{aligned} \tag{6.1}$$

6.1.1 Region 1 (S,S,S)

The first region corresponds to the worst-case bound defined by function $b_{3,1}$. We had the same intuition as for function $\phi_{2,1}$. For three constant step sizes, the worst-case function is given by replacing h by $h+h+h=3h$ in equations (4.4). Then, for three variable step sizes, we obtain our new worst-case function by replacing h by $h_1+h_2+h_3$. In other words, we only add the term $2h_3$ in the denominator of the first piece of function $\phi_{2,1}$. More precisely, the worst-case function is defined as follows:

$$\phi_{3,1}(x) = \begin{cases} \frac{LR}{2h_1+2h_2+2h_3+1}|x| - \frac{LR^2}{2(2h_1+2h_2+2h_3+1)^2}, & |x| \geq \frac{R}{2h_1+2h_2+2h_3+1} \\ \frac{L}{2}x^2, & |x| < \frac{R}{2h_1+2h_2+2h_3+1} \end{cases} \tag{6.2}$$

The behavior of the function remains the same as $\phi_{1,1}$ and $\phi_{2,1}$ as we observe in Figure 6.1. Only the slope changes, since we added a new term involving the third step size in the denominator. Here again, the interpolating function did not take the affine part at the right side of the solution.

We can easily prove that we reach the worst-case bound $\frac{LR^2}{2}b_{3,1}$ defined in equations (6.1) by generalizing the proof of function $\phi_{2,1}$ for three steps of GM. The first iterate is given by

$$x_1 = x_0 - \frac{h_1}{L}\phi'_{3,1}(x_0) = -R + \frac{Rh_1}{2(h_1+h_2+h_3)+1}$$

Then, the second iterate is given by

$$x_2 = x_1 - \frac{h_2}{L}\phi'_{3,1}(x_1) = -R + \frac{R(h_1+h_2)}{2(h_1+h_2+h_3)+1}$$

Finally, the third iterate is defined as:

$$x_N = x_3 = x_2 - \frac{h_3}{L}\phi'_{3,1}(x_2) = -R + \frac{R(h_1+h_2+h_3)}{2(h_1+h_2+h_3)+1}$$

Therefore, the objective function accuracy is expressed as:

$$\phi_{3,1}(x_N) - \phi_{3,1}(x^*) = \phi_{3,1}(x_N)$$

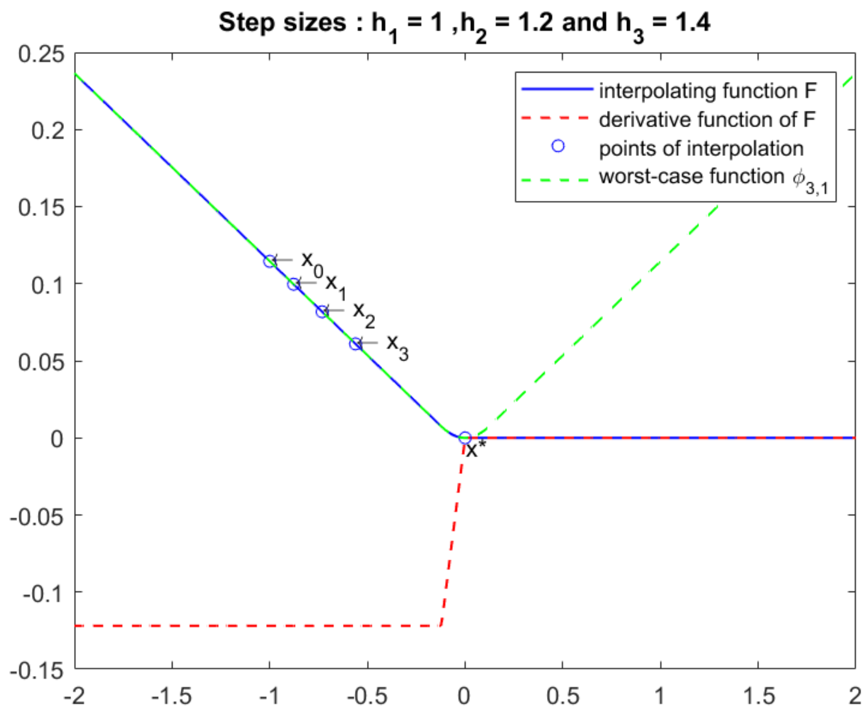


Figure 6.1: The iterates of the GM, the interpolating function and its derivative as well as the worst-case function $\phi_{3,1}$ for $(h_1, h_2, h_3) = (1, 1.2, 1.4)$ with the parameters L and R set to one.

$$
\begin{aligned}
&= \frac{-LR^2}{2(h_1+h_2+h_3)+1} - \frac{LR^2(h_1+h_2+h_3)}{(2(h_1+h_2+h_3)+1)^2} - \frac{LR^2}{2(2(h_1+h_2+h_3)+1)^2} \\
&= \frac{4LR^2(h_1+h_2+h_3)+2LR^2-2LR^2(h_1+h_2+h_3)-LR^2}{2(2(h_1+h_2+h_3)+1)^2} \\
&= \frac{LR^2}{2}\frac{1}{2(h_1+h_2+h_3)+1}
\end{aligned}
$$

Which is in fact equal to $\frac{LR^2}{2}b_{3,1}$.

6.1.2 Region 2 (S,B,B)

The second region corresponds to the worst-case bound $b_{3,2}$. Figure 6.2 shows the iterates and the interpolating function when $h_1 = 0.6$, $h_2 = 1.8$ and $h_3 = 1.8$. As expected, the first iterate stays in the affine part and does not oscillate around the optimal solution. That is the characteristic that we found for *Small* step sizes. Then, the second and third iterates, considered to be *Big*, overshoot the solution. We were inspired by the worst-case function of region (S, B) with two steps of GM. Firstly, we can suppose that the second piece of our function will be purely quadratic and equal to $\frac{Lx^2}{2}$. Then, here again, the aim is to find the slope of the affine part and where the change in the form of the derivative occurs. Also, as for $\phi_{2,4}$, we found out that the slope did not depend on step sizes h_2 and h_3. So, the first thing we tried was to take the same slope as for $\phi_{2,4}$. Since the second and third iterates are located in the quadratic part, our choice to test $\frac{-1}{h_1+1}$ for the slope is pertinent. The worst-case function is then defined as:

$$\phi^-_{3,2}(x) = \begin{cases} \frac{-LR}{1+h_1}x - \frac{LR^2}{2(1+h_1)^2}, & x \leq \frac{-R}{1+h_1} \\ \frac{L}{2}x^2, & x > \frac{-R}{1+h_1} \end{cases} \tag{6.3}$$

Figure 6.2 shows the worst-case function for $(h_1, h_2, h_3) = (0.6, 1.8, 1.8)$. It is not symmetric with the origin and therefore the function will depend on the sign of the starting iterate. The

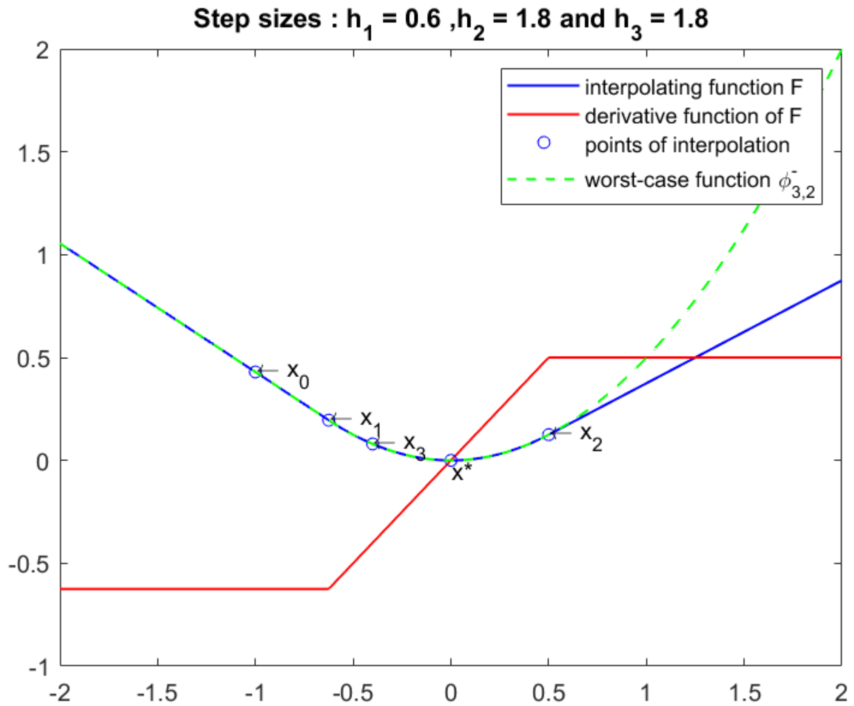


Figure 6.2: The iterates of the GM, the interpolating function and its derivative as well as the worst-case function $\phi^-_{3,2}$ for $(h_1, h_2, h_3) = (0.6, 1.8, 1.8)$ with the parameters L and R set to one.

proof is straightforward. We defined the three iterates:

$$\begin{aligned} x_1 &= x_0 - \frac{h_1}{L}\frac{d\phi^-_{3,2}}{dx}(x_0) \\ &= -R + \frac{Rh_1}{1+h_1} \\ &= \frac{-R}{1+h_1} \\ \text{and } x_2 &= x_1 - \frac{h_2}{L}\frac{d\phi^-_{3,2}}{dx}(x_1) \\ &= \frac{-R}{1+h_1} + h_2\frac{R}{1+h_1} \\ &= \frac{-R(1-h_2)}{1+h_1} \end{aligned}$$

$$\begin{aligned}\text{finally, } x_3 &= x_2 - \frac{h_3}{L}\frac{d\phi^-_{3,2}}{dx}(x_2)\\ &= \frac{-R(1-h_2)}{1+h_1} + h_3\frac{R(1-h_2)}{1+h_1}\\ &= \frac{-R(1-h_2)(1-h_3)}{1+h_1}\end{aligned}$$

Therefore,

$$\phi^-_{3,2}(x_N) = \frac{LR^2}{2}\frac{(1-h_2)^2(1-h_3)^2}{(1+h_1)^2}$$

Which is in fact equal to bound $\frac{LR^2}{2}b_{3,2}$.

6.1.3 Region 3 (S,S,B)

The third region corresponds to the worst-case bound $b_{3,3}$ defined in equations (6.1). The two first iterates are considered to be *Small* while the third is seen as *Big*. Figure 6.3 shows the iterates of the gradient method when $h_1 = 1$, $h_2 = 0.8$ and $h_3 = 2$ as well as the interpolating function associated. Here again, we can suppose that the second part of the worst-case function is equal to the usual quadratic form $\frac{Lx^2}{2}$. We now only need to find the slope of the affine part and where the change in the form of the derivative occurs. We found with our numeric resolutions that the slope does not depend on the third step size. Our intuition was to replace h_1 by $h_1 + h_2$ in the denominator of worst-case function $\phi_{3,2}$. We define thus the following worst-case function

$$\phi^-_{3,3}(x) = \begin{cases} \frac{-LR}{1+h_1+h_2}x - \frac{LR^2}{2(1+h_1+h_2)^2}, & x \le \frac{-R}{1+h_1+h_2} \\ \frac{L}{2}x^2, & x > \frac{-R}{1+h_1+h_2} \end{cases} \tag{6.4}$$

Figure 6.3 shows also the worst-case function that we just defined for $(h1_, h_2, h_3) = (1, 0.8, 2)$. As in the previous region, the function is not symmetric around the optimal solution, which makes it dependent on the sign of the starting point.

We can now demonstrate that the bound $\frac{LR^2}{2}b_{3,3}$ is achieved by function $\phi^-_{3,3}$ when applying three steps of GM. The iterates are given by

$$\begin{aligned} x_1 &= x_0 - \frac{h_1}{L}\frac{d\phi^-_{3,3}}{dx}(x_0)\\ &= -R + \frac{Rh_1}{1+h_1+h_2}\\ &= \frac{-R(1+h_2)}{1+h_1+h_2}\\ \text{and } x_2 &= x_1 - \frac{h_2}{L}\frac{d\phi^-_{3,3}}{dx}(x_1)\\ &= \frac{-R(1+h_2)}{1+h_1+h_2} + h_2\frac{R}{1+h_1+h_2}\\ &= \frac{-R}{1+h_1+h_2}\end{aligned}$$

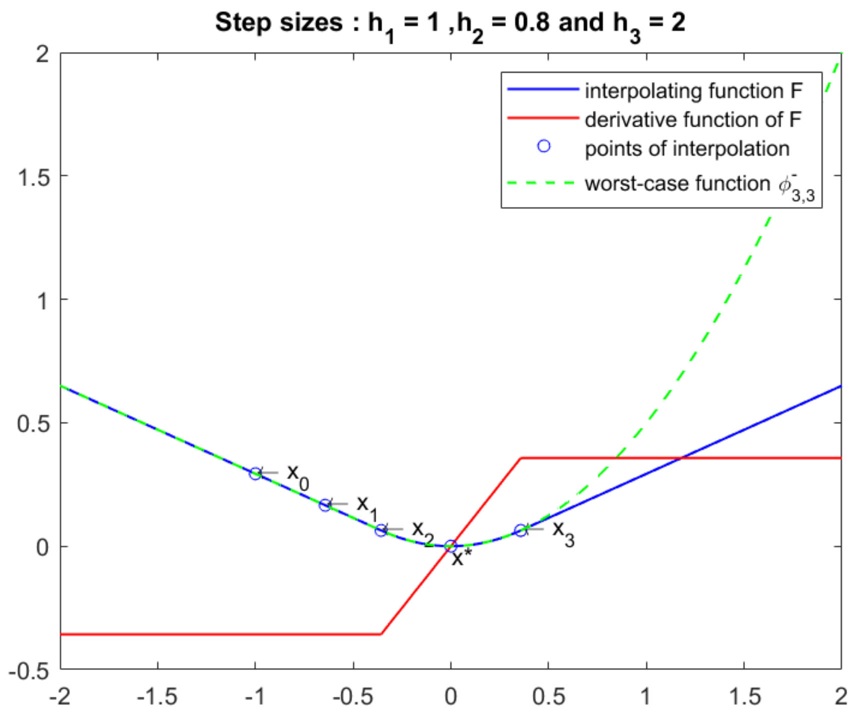


Figure 6.3: The iterates of the GM, the interpolating function and its derivative as well as the worst-case function $\phi^-_{3,3}$ for $(h_1, h_2, h_3) = (1, 0.8, 2)$ with the parameters L and R set to one.

$$\begin{aligned}\text{finally, } x_3 &= x_2 - \frac{h_3}{L}\frac{d\phi^-_{3,3}}{dx}(x_2)\\ &= \frac{-R}{1+h_1+h_3} + h_3\frac{R}{1+h_1+h_2}\\ &= \frac{-R(1-h_3)}{1+h_1+h_2}\end{aligned}$$

Therefore, the objective function accuracy is expressed as follows and reaches in fact the bound $\frac{LR^2}{2}b_{3,3}$:

$$\phi^-_{3,3}(x_N) = \frac{LR^2}{2}\frac{(1-h_3)^2}{(1+h_1+h_2)^2}$$

6.1.4 Region 4 (S,B,S)

The last region where h_1 is considered to be *Small* corresponds to the worst-case bound $b_{3,4}$ defined in equations (6.1). This is the most particular case, because we alternate between *Small* and *Big* step sizes. As can be seen in Figure 6.4, the first iterate did not oscillate around the optimal solution and stayed on the affine part of the interpolating function. Then, the second iterate overshoots the optimum, as the step size associated is considered to be *Big*. Since the third step size is also considered to be *Small*, it creates a second affine piece where the third iterate is located. The purely quadratic piece lies then between the two linear parts. The difference with the two previous worst-case functions is that the usual quadratic function $\frac{Lx^2}{2}$ is blocked by the affine pieces.

The purpose here is to find the values of the two slopes and where the two changes in the form of the derivative functions occurs. We found that the slope of the first linear part only depends on first step size h_1. We supposed that this piece function is equal to the first part of worst-case function $\phi^-_{3,2}$, characterized by a slope equal to $-1/(1+h_1)$. For the third piece, i.e., the second affine part, we noticed that the slope depends on the three step sizes. Our intuition was to be inspired by region (B, S). Firstly, we supposed that the slope m will have the following expression:

$$m = \eta(h_1)\frac{h_2 - 1}{2h_3 + 1}$$

where $\eta(h_1)$ will be the contribution in h_1. The term involving h_1 in $b_{3,4}$ is the same as in $b_{3,2}$. From this, we found that the contribution in h_1 will be equal to the opposite of the slope of the first affine part, i.e., $\eta(h_1) = 1/(h_1 + 1)$. The deduced worst-case function is defined as:

$$\phi^-_{3,4}(x) = \begin{cases} \frac{-LR}{1+h_1}x - \frac{-LR^2}{2(1+h_1)^2}, & x \leq \frac{-R}{1+h_1} \\ \frac{LR(h_2-1)}{(1+h_1)(2h_3+1)}x - \frac{LR^2}{2}\frac{(h_2-1)^2}{((1+h_1)(2h_3+1))^2}, & x \geq \frac{R(h_2-1)}{(1+h_1)(2h_3+1)} \\ \frac{L}{2}x^2, & \frac{-R}{1+h_1} < x < \frac{R(h_2-1)}{(1+h_1)(2h_3+1)} \end{cases} \tag{6.5}$$

Figure 6.4 shows the worst-case function just defined for $(h_1, h_2, h_3) = (1, 2, 0.4)$. This is the only case where the interpolating function matches perfectly the worst-case one everywhere. The previous interpolating functions always took a linear path instead of following the quadratic piece. The region (S,B,S) forces the interpolating function to take two affine parts created by the *Small* step sizes.

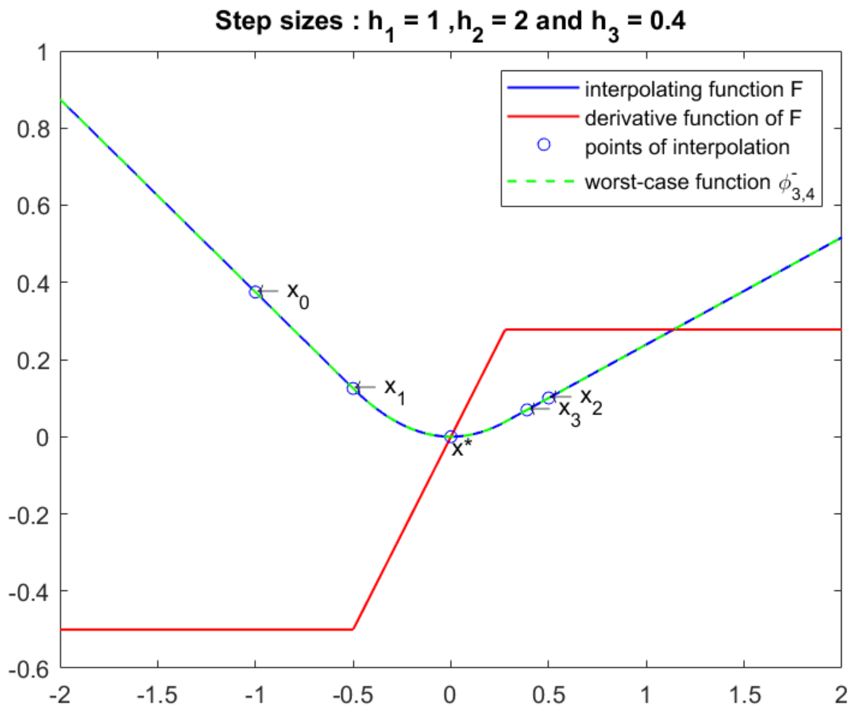


Figure 6.4: The iterates of the GM, the interpolating function and its derivative as well as the worst-case function $\phi^-_{3,4}$ for $(h_1, h_2, h_3) = (1, 2, 0.4)$ with the parameters L and R set to one.

Function $\phi_{3,4}^-$ is valid only for negative starting point. We now prove that the bound $\frac{LR^2}{2} b_{3,4}$ is achieved by function $\phi_{3,4}^-$ when applying three steps of GM. The three iterates are defined as:

$$\begin{aligned}
x_1 &= x_0 - \frac{h_1}{L}\frac{d\phi_{3,4}^-}{dx}(x_0) \\
&= -R + \frac{Rh_1}{1+h_1} \\
\text{and } x_2 &= x_1 - \frac{h_2}{L}\frac{d\phi_{3,4}^-}{dx}(x_1) \\
&= \frac{-R - Rh_1 + Rh_1 + Rh_2}{1+h_1} \\
&= \frac{R(h_2-1)}{1+h_1} \\
\text{finally, } x_N = x_3 &= x_2 - \frac{h_3}{L}\frac{d\phi_{3,4}^-}{dx}(x_2) \\
&= \frac{R(h_2-1)}{1+h_1} - \frac{h_3}{L}\frac{LR(h_2-1)}{(1+h_1)(2h_3+1)} \\
&= \frac{R(h_2-1)(h_3+1)}{(1+h_1)(2h_3+1)}
\end{aligned}$$

Therefore, the final objective function accuracy is given by

$$\begin{aligned}
\phi_{3,4}^-(x_N) &= \frac{LR^2(h_2-1)^2(h_3+1)}{((1+h_1)(2h_3+1))^2} - \frac{LR^2}{2}\frac{(h_2-1)^2}{((1+h_1)(2h_3+1))^2} \\
&= \frac{LR^2}{2}\left(\frac{1}{(1+h_1)^2}\right)(1-h_2)^2\left(\frac{1}{1+2h_3}\right)
\end{aligned}$$

Now that we have defined the worst-case function when h_1 is considered to be *Small*, we can look at the worst-case bounds when h_1 is considered to be *Big*. Unfortunately, we can find only three of the four worst-case bounds in [Dac19], as the last one not has been identified. The three bounds are defined as:

$$\begin{aligned}
b_{3,5}(h_1,h_2,h_3) &= \left(\frac{1}{1+2h_2+2h_3}\right)(1-h_1)^2 \\
b_{3,6}(h_1,h_2,h_3) &= \left(\frac{1}{(1+h_2)^2}\right)(1-h_1)^2(1-h_3)^2 \\
b_{3,7}(h_1,h_2,h_3) &= (1-h_1)^2(1-h_2)^2(1-h_3)^2
\end{aligned} \tag{6.6}$$

6.1.5 Region 5 (B,S,S)

The fifth region corresponds to the bound $b_{3,5}$. Figure 6.5 shows the iterates of GM as well as the interpolating function associated when $h_1 = 1.8$, $h = 0.6$ and $h_3 = 1$. The first step size is considered to be *Big* and therefore the first iterate overshoots the solution. The two other iterates stay on the same side and approach the solution by moving through the affine part of the function. As can be seen in Figure 6.5, we can imagine that the first piece of the worst-case function is purely quadratic and equal to $\frac{Lx^2}{2}$. The function will be quadratic until the derivative becomes constant. We were inspired by the region (B, S). Since we have one more iterate computed with a *Small* step size, we can imagine that the slope is obtained by adding $2h_3$ in the denominator of the linear piece of function $\phi^-_{2,3}$. We defined the following worst-case function:

$$\phi^-_{3,5}(x) = \begin{cases} \frac{LR(h_1-1)}{2h_2+2h_3+1}x - \frac{LR^2}{2}\left(\frac{h_1-1}{2h_2+2h_3+1}\right)^2, & x \geq \frac{R(h_1-1)}{2h_2+2h_3+1} \\ \frac{L}{2}x^2, & x < \frac{R(h_1-1)}{2h_2+2h_3+1} \end{cases} \tag{6.7}$$

We can observe the behavior of the worst-case function in Figure 6.5. As previously, we prove

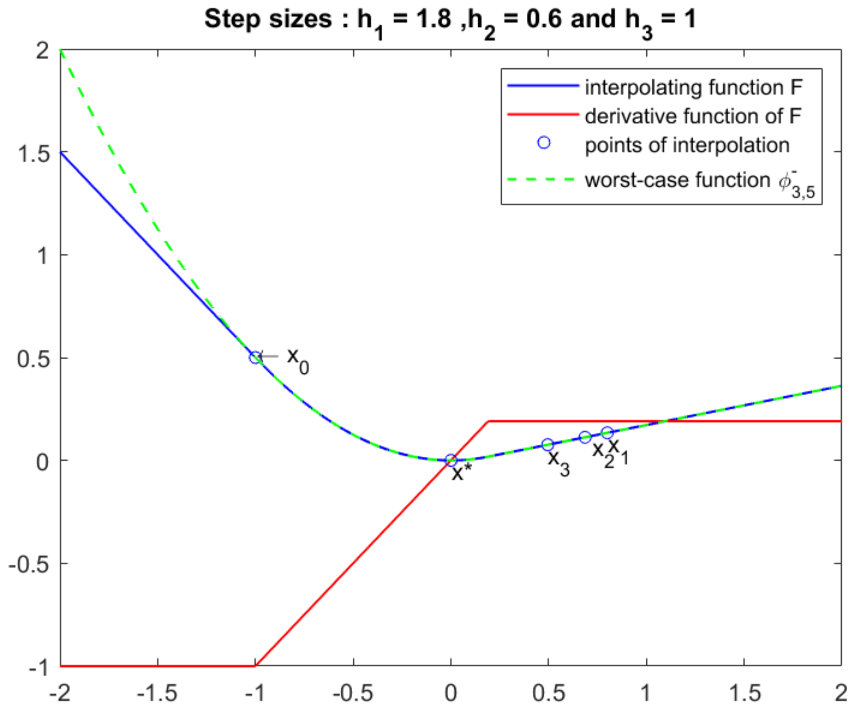


Figure 6.5: The iterates of the GM, the interpolating function and its derivative as well as the worst-case function $\phi^-_{3,5}$ for $(h_1, h_2, h_3) = (1.8, 0.6, 1)$ with the parameters L and R set to one.

that the bound $\frac{LR^2}{2}b_{3,5}$ is attained by three steps of GM on function $\phi^-_{3,5}$. The three iterates are defined as:

$$\begin{aligned} x_1 &= x_0 - \frac{h_1}{L}\frac{d\phi^-_{3,5}}{dx}(x_0) \\ &= R(h_1-1), \\ \text{and } x_2 &= x_1 - \frac{h_2}{L}\frac{d\phi^-_{3,5}}{dx}(x_1) \\ &= R(h_1-1) - h_2\frac{R(h_1-1)}{2(h_2+h_3)+1}) \\ &= \frac{R(h_1-1)(2(h_2+h_3)+1-h_2)}{2(h_2+h_3)+1}, \end{aligned}$$

$$\begin{aligned}\text{finally, } x_N = x_3 = x_2 - \frac{h_3}{L}\frac{d\phi^-_{3,5}}{dx}(x_2) \\ = \frac{R(h_1-1)(2(h_2+h_3)+1-h_2)}{2(h_2+h_3)+1} - h_3\frac{R(h_1-1)}{2(h_2+h_3)+1} \\ = \frac{R(h_1-1)(2(h_2+h_3)+1-h_2-h_3)}{2(h_2+h_3)+1}\end{aligned}$$

Therefore, the generalized objective function accuracy is expressed as:

$$\begin{aligned}\phi^-_{3,5}(x_N) &= \frac{LR^2(h_1-1)^2(2(h_2+h_3)+1-h_2-h_3)}{(2(h_2+h_3)+1)^2} - \frac{LR^2}{2}\frac{(h_1-1)^2}{(2(h_2+h_3)+1)^2} \\ &= \frac{LR^2(h_1-1)^2(4(h_2+h_3)+2-2h_2-2h_3-1)}{2(2(h_2+h_3)+1)^2} \\ &= \frac{LR^2}{2}\frac{1}{2h_2+2h_3+1}(h_1-1)^2\end{aligned}$$

Which is indeed equal to $\frac{LR^2}{2}b_{3,5}$

6.1.6 Region 6 (B,S,B)

The sixth region corresponds to the worst-case bound $b_{3,6}$. Figure 6.6 shows the different iterates of GM and the interpolating function associated for step sizes $(h_1, h_2, h_3) = (1.8, 1, 1.8)$. It is obvious now, that the first iterate overshoots the solution, as the corresponding step size is considered to be *Big*. The second iterate does not oscillate around the optimum and stays in the affine part while approaching the solution. Finally, the third iterate again overshoots the solution. The first piece of the function will be equal to the classic quadratic form $\frac{Lx^2}{2}$ until we reach the affine part of the function. The purpose is then to find the value of the slope and where the derivative becomes constant. Firstly, we found that the slope does not depend on third step size h_3. As the last iterate is located on the left side of the solution (i.e., in the quadratic piece), the behavior of the combination of step sizes h_1 and h_2 is similar to the one of region (B, S). We can then suppose that the slope is the same, i.e., equal to $(h_1-1)/(2h_2+1)$. Therefore, we define the following worst-case function:

$$\phi^-_{3,6}(x) = \begin{cases} \frac{LR(h_1-1)}{1+h_2}x - \frac{LR^2(h_1-1)^2}{2(1+h_2)^2}, & x \geq \frac{R(h_1-1)}{1+h_2} \\ \frac{L}{2}x^2, & x < \frac{h_1-1}{1+h_1} \end{cases} \tag{6.8}$$

We apply now three steps of GM on the function defined above. The three iterates are defined as follows:

$$\begin{aligned}x_1 &= x_0 - \frac{h_1}{L}\frac{d\phi^-_{3,6}}{dx}(x_0) \\ &= R(h_1-1), \\ \text{and } x_2 &= x_1 - \frac{h_2}{L}\frac{d\phi^-_{3,6}}{dx}(x_1) \\ &= R(h_1-1) - h_2\frac{R(h_1-1)}{h_2+1}\end{aligned}$$

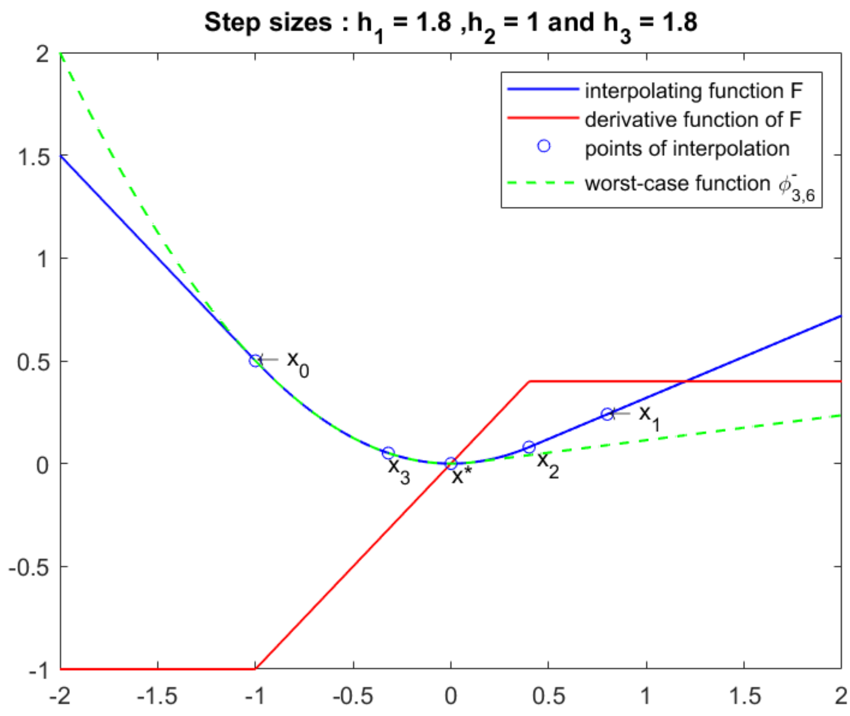


Figure 6.6: The iterates of the GM, the interpolating function and his derivative as well as the worst-case function $\phi^-_{3,6}$ for $(h_1, h_2, h_3) = (1.8, 1, 1.8)$ with the parameters L and R set to one.

$$= \frac{R(h_1-1)}{h_2+1},$$

$$\text{finally, } x_N = x_3 = x_2 - \frac{h_3}{L}\frac{d\phi_{3,6}}{dx}(x_2)$$
$$= \frac{R(h_1-1)}{h_2+1} - h_3\frac{R(h_1-1)}{h_2+1}$$
$$= \frac{R(h_1-1)(1-h_3)}{h_2+1}$$

Therefore, the objective function accuracy is defined as follows:

$$\phi_{3,6}(x_N) = \frac{LR^2}{2}\left(\frac{1}{(1+h_2)^2}\right)(h_1-1)^2(1-h_3)^2$$

Which is indeed equal to $\frac{LR^2}{2}b_{3,6}$.

6.1.7 Region 7 (B,B,B)

The region corresponding to three *Big* step sizes is characterized by the bound $b_{3,7}$. This bound is the same as bound $b_{2,2}$ multiplied by $(1-h_3)^2$. Since the worst-case function corresponding to the bound $b_{2,2}$ does not depend on the step sizes, we can suppose that the worst-case function for three *Big* step sizes is also equal to $\frac{Lx^2}{2}$. Therefore,

$$\phi_{3,7}(x) = \frac{Lx^2}{2} \tag{6.9}$$

We can see in Figure 6.7 the interpolating function, the iterates and the worst-case function when $L = R = 1$ and for $(h_1, h_2, h_3) = (1.8, 1.6, 1.9)$. The proof is straightforward. Indeed, the

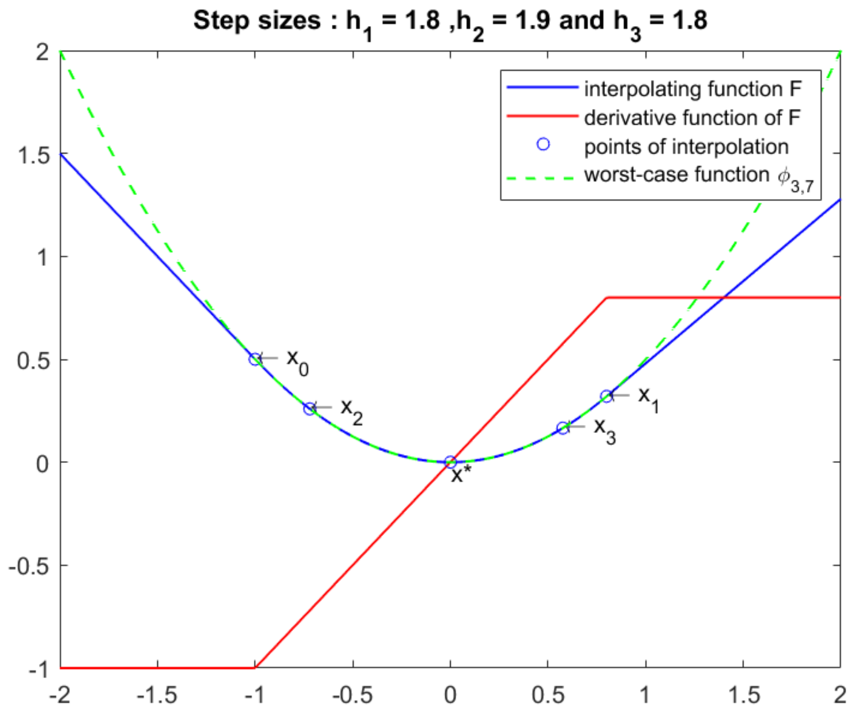


Figure 6.7: The iterates of the GM, the interpolating function and its derivative as well as the worst-case function $\phi^-_{3,7}$ for $(h_1, h_2, h_3) = (1.8, 1, 1.8)$ with the parameters L and R set to one.

iterates are defined as:

$$\begin{aligned} x_1 &= x_0 - \frac{h_1}{L}\frac{d\phi_{3,7}}{dx}(x_0) \\ &= R(h_1 - 1) \\ \text{and } x_2 &= x_1 - \frac{h_2}{L}\frac{d\phi_{3,7}}{dx}(x_1) \\ &= x_1(1 - h_2) \\ &= -R(1 - h_1)(1 - h_2) \\ \text{finally, } x_N = x_3 &= x_2 - \frac{h_3}{L}\frac{d\phi_{3,7}}{dx}(x_2) \\ &= -R(1 - h_1)(1 - h_2)(1 - h_3) \end{aligned}$$

Therefore,

$$\phi_{3,7}(x_N) - \phi_{3,7}(x^*) = \phi_{3,7}(x_N) = \frac{LR^2}{2}(1 - h_1)^2(1 - h_2)^2(1 - h_3)^2$$

6.1.8 Region 8 (B,B,S)

Until now, there has been no derived bound for three steps of GM with step sizes belonging to the regime (B, B, S). As explained previously, there were some combinations of step sizes for which the sdp-PEP problems associated were not one-dimensional. More precisely, we found that the solutions of (sdp-PEP) for the gradient method with such step sizes are two-dimensional. Since the interpolation code is tailored only for one-dimensional inputs, we were not able to compute the required interpolating functions.

However, we have gathered some information about the worst-case function by analyzing the set of triples $\{(x_i, f_i, g_i)\}_{i\in\{0,1,2,3,*\}}$. Firstly, we found with numerical resolutions that the gradients at second and third iterates are equal, i.e., $\nabla F(x_2, y_2) = \nabla F(x_3, y_3)$. Also, for $(x_0, y_0) = (-1, 0)$, we have that

$$\left.\frac{\partial F}{\partial x}\right|_{x_2} = \left.\frac{\partial F}{\partial x}\right|_{x_3} \leq 0 \text{ and } \left.\frac{\partial F}{\partial y}\right|_{y_2} = \left.\frac{\partial F}{\partial y}\right|_{y_3} \geq 0$$

We also found that $x_2 \leq x_3 \leq 0$ and $y_2 \geq y_3 \geq 0$. From this, we can imagine that the worst-case function between second and third iterates takes the form of a plane. We found another useful information about iterates. It seems that $x_1 = h_1 - 1$, as for regions 5, 6 and 7. Since $f_1 = 0.5$, we can imagine that the first piece has the following expression:

$$\phi^1_{2,8}(x, y) = \frac{x^2}{2} + \eta(y)$$

Where $\eta(y)$ is the contribution in y. For example, the resolution of (sdp-PEP) for $(h_1, h_2, h_3) = (1.9, 1.8, 0.8)$, gives the following set of points

Step	(x_i, y_i)	f_i	$\nabla f(x_i, y_i)$
0	$(-1, 0)$	0.5	$(-1, 1e-06)$
1	$(0.9, 1e-08)$	0.30	$(0.64, -0.36)$
2	$(-0.26, 0.65)$	0.14	$(-0.15, 0.2)$
3	$(-0.14, 0.49)$	0.09	$(-0.15, 0.2)$
$*$	$(0, 0)$	0	$(0, 0)$

We believe that by being able to compute the interpolation functions, we will be able to deduce the worst-case function and subsequently identify the missing worst-case bound.

6.2 Conclusion

For three steps of GM, there is still no conjecture in the literature for a tight worst-case bound on the objective function accuracy. As explained above, a possible conjecture will be expressed as the maximum of eight functions that can be seen as worst-case bounds attained by eight different types of combination of step sizes. Seven of the eight functions can be found in [Dac19]. Therefore, we identified seven worst-case functions. Each of these functions reaches one of the seven bounds. These functions are summarized in Table 6.1. The eighth function that composes the worst-case bound corresponds to the regime (B, B, S). We found that the solutions of problem (sdp-PEP) according to function accuracy with step sizes belonging to the (B, B, S) regime, are two-dimensional. We have not been able to identify the last worst-case function because the interpolation code is calibrated for only one-dimensional inputs. However, it may be easier to deduce the worst-case function first and then derive the worst-case bound associated.

	$\mu = 0$
(S,S,S)	$\phi_{3,1}(x) = \begin{cases} \frac{LR}{2h_1+2h_2+2h_3+1}\lvert x\rvert - \frac{LR^2}{2(2h_1+2h_2+2h_3+1)^2}, & \lvert x\rvert \geq \frac{R}{2h_1+2h_2+2h_3+1} \\ \frac{L}{2}x^2, & \lvert x\rvert < \frac{R}{2h_1+2h_2+2h_3+1} \end{cases}$
(S,B,B)	$\phi^-_{3,2}(x) = \begin{cases} \frac{-LR}{1+h_1}x - \frac{LR^2}{2(1+h_1)^2}, & x \leq \frac{-R}{1+h_1} \\ \frac{L}{2}x^2, & x > \frac{-R}{1+h_1} \end{cases}$
(S,S,B)	$\phi^-_{3,3}(x) = \begin{cases} \frac{-LR}{1+h_1+h_2}x - \frac{LR^2}{2(1+h_1+h_2)^2}, & x \leq \frac{-R}{1+h_1+h_2} \\ \frac{L}{2}x^2, & x > \frac{-R}{1+h_1+h_2} \end{cases}$
(S,B,S)	$\phi^-_{3,4}(x) = \begin{cases} \frac{-LR}{1+h_1}x - \frac{-LR^2}{2(1+h_1)^2}, & x \leq \frac{-R}{1+h_1} \\ \frac{LR(h_2-1)}{(1+h_1)(2h_3+1)}x - \frac{LR^2}{2}\frac{(h_2-1)^2}{((1+h_1)(2h_3+1))^2}, & x \geq \frac{R(h_2-1)}{(1+h_1)(2h_3+1)} \\ \frac{L}{2}x^2, & \frac{-R}{1+h_1} < x < \frac{R(h_2-1)}{(1+h_1)(2h_3+1)} \end{cases}$
(B,S,S)	$\phi^-_{3,5}(x) = \begin{cases} \frac{LR(h_1-1)}{2h_2+2h_3+1}x - \frac{LR^2}{2}\left(\frac{h_1-1}{2h_2+2h_3+1}\right)^2, & x \geq \frac{R(h_1-1)}{2h_2+2h_3+1} \\ \frac{L}{2}x^2, & x < \frac{R(h_1-1)}{2h_2+2h_3+1} \end{cases}$
(B,S,B)	$\phi^-_{3,6}(x) = \begin{cases} \frac{LR(h_1-1)}{1+h_2}x - \frac{LR^2(h_1-1)^2}{2(1+h_2)^2}, & x \geq \frac{R(h_1-1)}{1+h_2} \\ \frac{L}{2}x^2, & x < \frac{h_1-1}{1+h_1} \end{cases}$
(B,B,B)	$\phi_{3,7}(x) = \frac{Lx^2}{2}$

Table 6.1: Worst-case functions for three steps of GM on smooth convex functions.

Chapter 7

N steps of gradient method

As we have seen in the previous chapters, it is possible to find some conclusions on the worst-case functions for N steps of GM with variables fixed step sizes. As detailed in [Dac19, Section 4.4.2], a conjecture for the worst-case bound on the objective function accuracy of N steps of GM, will be expressed as the maximum of 2^N functions. Therefore, we admits that there are at least 2^N worst-case functions. For each functional class, we can find in [Dac19] the generalized worst-case bound for three types of combination between the step sizes. We begin with the case of L-smooth convex functions.

7.1 Smooth convex functions

Even if we have not been able to identify the eight worst-case functions for three steps of GM, we can generalize the worst-case functions for N steps of GM for some types of combinations. Indeed, we can find in [Dac19] a few worst-case bounds on the objective function accuracy for N steps of GM and for specific regimes. Firstly, we can find the following generalization of the worst-case bound for the regime $(S, S, \ldots, S)$ or S^N:

$$b_{S^N}(\mathbf{h}) = \frac{1}{1 + \sum_{i=1}^{N} 2h_i} \tag{7.1}$$

This bound is valid for *Small* values of h_i. From this, we can derive the worst-case function associated:

$$\phi_{N,S}^{R,L}(x) = \begin{cases} \frac{LR}{1+\sum_{i=1}^{N} 2h_i}|x| - \frac{LR^2}{2(1+\sum_{i=1}^{N} 2h_i)^2}, & |x| \geq \frac{R}{1+\sum_{i=1}^{N} 2h_i} \\ \frac{L}{2}x^2, & |x| < \frac{R}{1+\sum_{i=1}^{N} 2h_i} \end{cases}. \tag{7.2}$$

The proof is straightforward using the proofs for region (S, S) and (S, S, S). We have for the first iterate,

$$x_1 = x_0 - \frac{h_1}{L}\frac{d\phi_{N,S}^{R,L}}{dx}(x_0) = -R + \frac{h_1}{L}\frac{LR}{\sum_{i=1}^{N} 2h_i + 1} = \frac{-R(1 + h_1 + \sum_{i=2}^{N} 2h_i)}{1 + \sum_{i=1}^{N} 2h_i}$$

Therefore, any iterate x_j is given by

$$x_j = x_{j-1} - \frac{h_j}{L}\frac{d\phi_{N,S}^{R,L}}{dx}(x_{j-1}) = \frac{-R(1 + \sum_{i=1}^{j} h_i + \sum_{i=j+1}^{N} 2h_i)}{1 + \sum_{i=1}^{N} 2h_i}$$

Also, the last iterate is defined as follows:

$$x_N = \frac{-R(1+\sum_{i=1}^N h_i)}{1+\sum_{i=1}^N 2h_i}$$

Finally, the objective function accuracy is given by

$$\begin{aligned}
\phi_{N,S}^{R,L}(x_N) - \phi_{N,S}^{R,L}(x^*) &= \phi_{N,S}^{R,L}(x_N) \\
&= LR^2 \frac{(1+\sum_{i=1}^N h_i)}{(1+\sum_{i=1}^N 2h_i)^2} - \frac{LR^2}{2(1+\sum_{i=1}^N 2h_i)^2} \\
&= \frac{LR^2(1+\sum_{i=1}^N 2h_i)}{2(1+\sum_{i=1}^N 2h_i)^2} \\
&= \frac{LR^2}{2} \frac{1}{1+\sum_{i=1}^N 2h_i}
\end{aligned}$$

Which is in fact equal to $\frac{LR^2}{2} b_{S^N}(\mathbf{h})$. Figure 7.1 shows an example for seven iterations and with the vector $\mathbf{h} = (0.9, 0.7, 1.3, 1, 0.8, 1, 1.2)$. The same observations can be made as for one, two and three steps of GM in the region S^N. The iterates stay in the first linear part of the function while approaching the optimal solution. However, they never come close to the optimum.

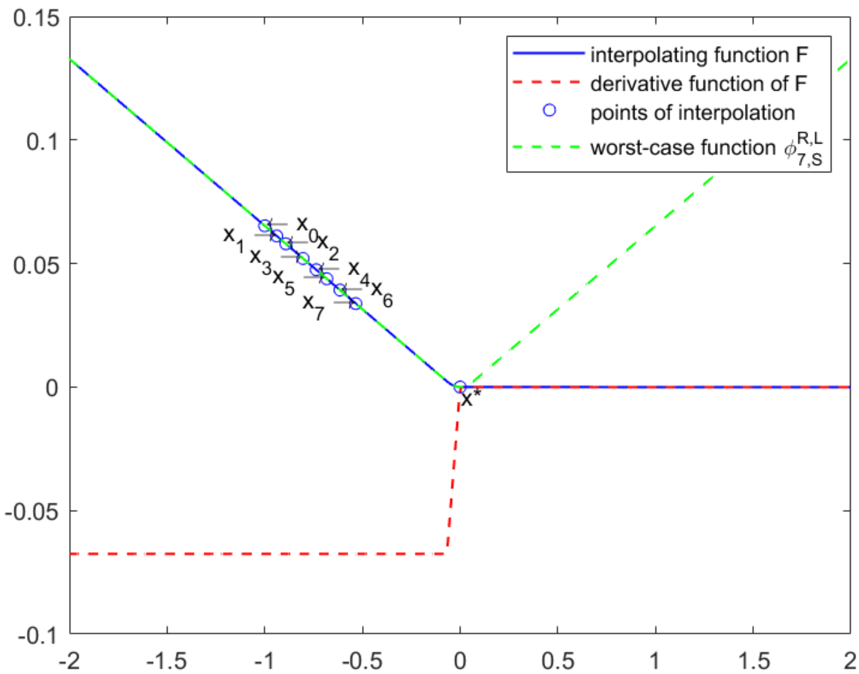


Figure 7.1: The iterates, the interpolating function and its derivative as well as the worst-case function associated for seven iterations of GM with $\mathbf{h} = (0.9, 0.7, 1.3, 1, 0.8, 1, 1.2)$.

We can also find a tight worst-case bound for N steps of GM for step sizes that belong to the regime $(B, B, \dots, B)$, denoted B^N. This bound is only valid for *Big* step sizes and is defined as follows:

$$b_{B^N}(\mathbf{h}) = \prod_{i=1}^{N}(1-h_i)^2 \tag{7.3}$$

The worst-case function associated is given by the well-known quadratic form:

$$\phi_{N,B}^{R,L} = L\frac{x^2}{2} \tag{7.4}$$

Figure 7.2 shows an example of interpolating function, iterates and worst-case function associated for $N = 10$ and with vector $\mathbf{h}$ defined as $linspace(1.89, 2, 10)$. The proof is also straightforward. The N^{th} iterate is given by

$$x_N = -\prod_{i=1}^{N}(1-h_i)$$

This expression shows that in fact, each iterate overshoots the optimal solution and changes of sign. Finally, the objective function accuracy after N steps is defined as follows:

$$\phi_{N,B}^{R,L}(x_N) = \frac{1}{2}\prod_{i=1}^{N}(1-h_i)^2$$

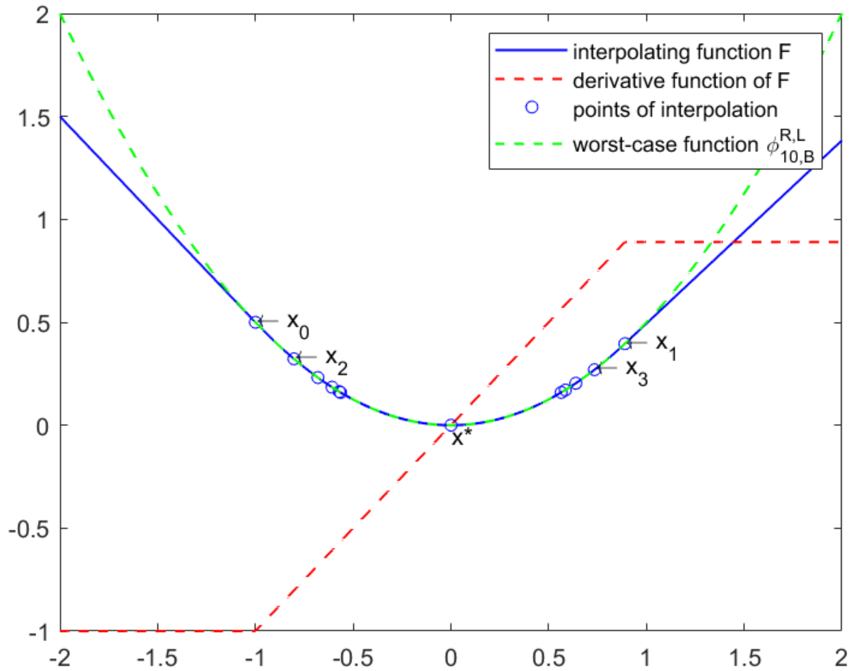


Figure 7.2: The iterates, the interpolating function and its derivative as well as the worst-case function associated for seven iterations of GM with $\mathbf{h} = linspace(1.89, 2, 10)$.

The last tight worst-case bound that we can find in [Dac19] for N steps of GM is the following:

$$b_{S^{N-1},B} = \frac{1}{(1+\sum_{i=1}^{N-1}h_i)^2}(1-h_N)^2 \tag{7.5}$$

This bound is valid for a combination of step sizes where all h_i are considered to be *Small* except the last one which is considered to be *Big*. The worst-case function associated is defined as follows

$$\phi^{R,L,-}_{NS,B}(x) = \begin{cases} \frac{-LR}{1+\sum_{i=1}^{N-1} h_i}x - \frac{LR^2}{2(1+\sum_{i=1}^{N-1} h_i)^2}, & x \leq \frac{-R}{1+\sum_{i=1}^{N-1} h_i} \\ \frac{L}{2}x^2, & x > \frac{-R}{1+\sum_{i=1}^{N-1} h_i} \end{cases}. \quad (7.6)$$

Figure 7.3 shows the interpolating function, the iterates and the worst-case function for $N = 10$ and with $\mathbf{h} = (0.2, 0.9, 0.2, 0.3, 2)$. We can make the same observations as for region (S, S, B). The iterates computed with a *Small* step size stay in the first piece of the function, i.e., the affine part. Then, the last iterate computed with a *Big* step size overshoots the optimal solution.

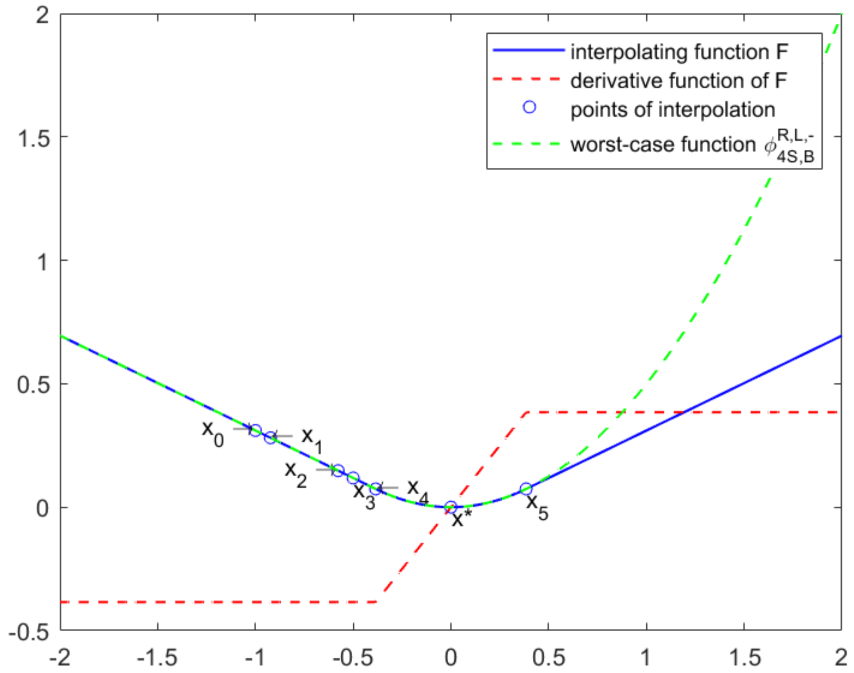


Figure 7.3: The iterates, the interpolating function and its derivative as well as the worst-case function associated for seven iterations of GM with $\mathbf{h} = (0.2, 0.9, 0.2, 0.3, 2)$.

Let's prove that the bound $\frac{LR^2}{2}b_{S^{N-1},B}$ is attained by N steps of GM on function $\phi^{R,L,-}_{NS,B}$. Since the latter function is not symmetric with the y-axis, function $\phi^{R,L,-}_{NS,B}$ is only valid for negative starting points. Therefore, we set $x_0 = -R$. The first iterate is given by

$$x_1 = x_0 - \frac{h_1}{L}\frac{d\phi^{R,L,-}_{NS,B}}{dx}(x_0) = -R + \frac{Rh_1}{1+\sum_{i=1}^{N-1} h_i} = \frac{-R(1+\sum_{i=2}^{N-1} h_i)}{1+\sum_{i=1}^{N-1} h_i}$$

The second iterate is given by

$$x_2 = x_1 - \frac{h_2}{L}\frac{d\phi^{R,L,-}_{NS,B}}{dx}(x_1)$$

$$
\begin{aligned}
&= \frac{-R(1+\sum_{i=2}^{N-1} h_i)}{1+\sum_{i=1}^{N-1} h_i} + h_2 \frac{R}{1+\sum_{i=1}^{N-1} h_i} \\
&= \frac{-R(1+\sum_{i=3}^{N-1} h_i)}{1+\sum_{i=1}^{N-1} h_i}
\end{aligned}
$$

Then, the $(N-1)^{th}$ iterate is given by

$$
x_{N-1} = \frac{-R}{1+\sum_{i=1}^{N-1} h_i}
$$

Finally, the last iterate is defined as follows:

$$
\begin{aligned}
x_N &= x_{N-1} - \frac{h_N}{L} \frac{d\phi_{NS,B}^{R,L,-}}{dx}(x_{N-1}) \\
&= \frac{-R}{1+\sum_{i=1}^{N-1} h_i} + h_N \frac{R}{1+\sum_{i=1}^{N-1} h_i} \\
&= \frac{-R(1-h_N)}{1+\sum_{i=1}^{N-1} h_i}
\end{aligned}
$$

Therefore, the objective final accuracy is equal to

$$
\phi_{NS,B}^{R,L,-}(x_N) = \frac{LR^2}{2} \frac{(1-h_N)^2}{(1+\sum_{i=1}^{N-1} h_i)^2}
$$

Which is in fact equal to $\frac{LR^2}{2} b_{S^{N-1},B}$.

7.2 Smooth strongly convex functions

For N steps of the gradient method applied to smooth strongly convex functions, we can also generalize the worst-case functions according to specific regimes. We can find in [Dac19] the following tight worst-case bound for the objective function accuracy for step sizes that belong to the regime $(S, S, \ldots, S)$ or S^N:

$$
b_{S^N}^{\mu}(\mathbf{h}) = \frac{\kappa}{(\kappa-1) + \prod_{i=1}^{N} (1-\kappa h_i)^{-2}}
$$

This bound is valid for *Small* values of h_i. From function $\phi_{2,1}^{\mu}$, we can derive the following worst-case function associated to the regime S^N:

$$
\phi_{N,S}^{\mu}(x) = \begin{cases} \frac{\mu}{2}x^2 + a_\tau |x| + b_\tau & \text{if } |x| \geq \tau \\ \frac{L}{2}x^2 & \text{else} \end{cases} \tag{7.7}
$$

where $a_\tau = (L-\mu)\tau$, $b_\tau = -\left(\frac{L-\mu}{2}\right)\tau^2$ and

$$
\tau = \frac{R\kappa}{(\kappa-1) + \prod_{i=1}^{N} (1-\kappa h_i)^{-2}}
$$

We can prove that the latter function reaches the bound $\frac{LR^2}{2}b^{\mu}_{S^N}$ when applying N steps of GM. The N^{th} iterate is defined as follows:

$$\begin{aligned} x_N &= x_{N-1} - \frac{h_N}{L}\frac{d\phi^{\mu}_{N,S}}{dx}(x_{N-1}) \\ &= \frac{-R\left((\kappa-1)+\prod_{i=1}^{N}(1-\kappa h_i)^{-1}\right)}{(\kappa-1)+\prod_{i=1}^{N}(1-\kappa h_i)^{-2}} \end{aligned}$$

Therefore, the objective function accuracy after N steps is given by

$$\begin{aligned} \phi^{\mu}_{N,S}(x_N) - \phi^{\mu}_{N,S}(x_*) &= \phi^{\mu}_{N,S}(x_N) \\ &= \frac{\mu}{2}\left(\frac{-R\left((\kappa-1)+\prod_{i=1}^{N}(1-\kappa h_i)^{-1}\right)}{(\kappa-1)+\prod_{i=1}^{N}(1-\kappa h_i)^{-2}}\right)^2 \\ &\quad + \frac{R^2\mu(1-\kappa)\left((\kappa-1)+\prod_{i=1}^{N}(1-\kappa h_i)^{-1}\right)}{\left[(\kappa-1)+\prod_{i=1}^{N}(1-\kappa h_i)^{-2}\right]^2} \\ &\quad - \frac{R^2\frac{\mu^2}{L}(1-\kappa)}{2\left[(\kappa-1)+\prod_{i=1}^{N}(1-\kappa h_i)^{-2}\right]^2} \\ &= \frac{R^2\mu(\kappa-1)^2 + R^2\mu\prod_{i=1}^{N}(1-\kappa h_i)^{-2} + 2R^2\mu(\kappa-1)\prod_{i=1}^{N}(1-\kappa h_i)^{-1}}{2\left[(\kappa-1)+\prod_{i=1}^{N}(1-\kappa h_i)^{-2}\right]^2} \\ &\quad - \frac{2R^2\mu(\kappa-1)\left((\kappa-1)+\prod_{i=1}^{N}(1-\kappa h_i)^{-1}\right)}{2\left[(\kappa-1)+\prod_{i=1}^{N}(1-\kappa h_i)^{-2}\right]^2} \\ &\quad - \frac{R^2\frac{\mu^2}{L}(1-\kappa)}{2\left[(\kappa-1)+\prod_{i=1}^{N}(1-\kappa h_i)^{-2}\right]^2} \\ &= \frac{R^2(\kappa-1)(-\mu\kappa+\mu+\mu\kappa) + R^2\mu\prod_{i=1}^{N}(1-\kappa h_i)^{-2}}{2\left[(\kappa-1)+\prod_{i=1}^{N}(1-\kappa h_i)^{-2}\right]^2} \\ &= \frac{R^2\mu\left((\kappa-1)+\prod_{i=1}^{N}(1-\kappa h_i)^{-2}\right)}{2\left[(\kappa-1)+\prod_{i=1}^{N}(1-\kappa h_i)^{-2}\right]^2} \\ &= \frac{LR^2}{2}\frac{\kappa}{(\kappa-1)+\prod_{i=1}^{N}(1-\kappa h_i)^{-2}} \end{aligned}$$

This last expression is in fact, equal to $\frac{LR^2}{2}b^{\mu}_{S^N}$.

The tight worst-case bound associated to the regime B^N for smooth strongly convex functions is the same as for the smooth convex case. More precisely,

$$b^{\mu}_{B^N}(\mathbf{h}) = b_{B^N}(\mathbf{h}) = \prod_{i=1}^{N}(1-h_i)^2$$

Therefore, the worst-case function is also the same:

$$\phi^{\mu}_{N,B} = \phi^{R,L}_{N,B} = \frac{Lx^2}{2}$$

The worst-case behavior of GM according to objective function accuracy for the regime (S_{N-1}, B) is given by the following function:

$$b^{\mu}_{S^{N-1},B}(\mathbf{h}) = \left(\frac{\kappa}{(\kappa-1) + \prod_{i=1}^{N-1}(1-\kappa h_i)^{-1}}\right)^2 (1-h_N)^2$$

This bound is valid for step sizes such that all but the last one are considered *Small*. Then, the worst-case function associated is defined as follows:

$$\phi^{\mu,-}_{NS,B}(x) = \begin{cases} \frac{\mu}{2}x^2 - a_\tau x + b_\tau & \text{if } x \leq -\tau \\ \frac{L}{2}x^2 & \text{else} \end{cases} \tag{7.8}$$

where $a_\tau = (L-\mu)\tau$, $b_\tau = -\left(\frac{L-\mu}{2}\right)\tau^2$ and

$$\tau = \frac{R\kappa}{(\kappa-1) + \prod_{i=1}^{N-1}(1-\kappa h_i)^{-1}}$$

We can finally prove that the bound $\frac{LR^2}{2}b^{\mu}_{S^{N-1},B}$ is reached by N steps of GM applied to the function just defined. The $(N-1)^{th}$ iterate is given by

$$\begin{aligned} x_{N-1} &= x_{N-2} - \frac{h_{N-1}}{L}\frac{d\phi^{\mu,-}_{NS,B}}{dx}(x_{N-2}) \\ &= -\tau \end{aligned}$$

Then, the last iterate x_N is defined as:

$$\begin{aligned} x_N &= x_{N-1} - \frac{h_N}{L}\frac{d\phi^{\mu,-}_{NS,B}}{dx}(x_{N-1}) \\ &= -\tau - \frac{h_N}{L}(-L\tau) \\ &= -\tau(1-h_N) \end{aligned}$$

Finally, the objective function accuracy is given by

$$\begin{aligned} \phi^{\mu,-}_{NS,B}(x_N) - \phi^{\mu,-}_{NS,B}(x^*) &= \phi^{\mu,-}_{NS,B}(x_N) \\ &= \frac{L\tau^2}{2}(1-h_N)^2 \\ &= \frac{LR^2}{2}\left(\frac{\kappa}{(\kappa-1) + \prod_{i=1}^{N-1}(1-\kappa h_i)^{-1}}\right)^2 (1-h_N)^2 \end{aligned}$$

This last expression is in fact, equal to $\frac{LR^2}{2}b^{\mu}_{S^{N-1},B}$.

7.3 Conclusion

Based on the number of functions composing the worst-case bound for N iterations and on our results for two and three steps, we acknowledge that there are also at least 2^N worst-case functions. It was also found that for the majority of cases, the worst-case function depends on the sign of the starting iterate. In this master thesis, we focused only on functions valid for any negative starting points provided that $\|x_0 - x_*\| \leq R$. Unfortunately, it is apparently complicated to derive the worst-case functions as the number of iterations increases. For example, we observed that for some combinations of step sizes the performance estimation problems were three-dimensional for five iterations or five-dimensional for eight iterations. However, we were able to identify the worst-case functions associated with specific regimes for N steps of GM for both functional classes.

Chapter 8

Conclusion

8.1 Research outcomes

This master thesis tries to bring new elements in the analysis of worst-case performances of first-order optimization methods. We focused our work on the gradient method with fixed variable step sizes. More precisely, the aim of this thesis was to derive the functions that achieve the worst-case behaviors of the gradient method according to the objective function accuracy.

We can find in the literature conjectures on the behavior of the gradient method with fixed constant step sizes for the objective function accuracy $f(x_N) - f(x_*)$. Conjecture (4.1.), derived in [DT14], provides us with a worst-case bound for smooth convex unconstrained minimization. This conjecture is extended to the class of smooth strongly functions in [Tay17] and corresponds to Conjecture (4.2.) in this work. Moreover, it has been found that the worst-case behavior of the gradient method according to objective function accuracy is achieved by a one-dimensional function for both functional classes.

One step of gradient method. The worst-case bounds of Conjectures (4.1.) and (4.2.) for one iteration of gradient method are both expressed as the maximum of two functions depending on unique step size h. We have seen that the optimal step size (optimal in the sense of achieving the lowest worst-case) for one step of gradient method is given by $h_{opt} = 1.5$. Therefore, when $h \leq h_{opt}$, the worst-case bound is given by the first part of max expression in Conjectures (4.1.) and (4.2.), depending on whether the objective function belongs to the class of smooth convex or smooth strongly convex functions. Otherwise, when $h > h_{opt}$, the worst-case behavior according to objective function accuracy is given by the second part of previous conjectures.

The worst-case functions that were already known for N constant step sizes, were numerically verified for one step in this work. The second parts of max expression in Conjectures (4.1.) and (4.2.) are attained by the same purely quadratic function. The first bound that composes Conjecture (4.1.) is attained by one step of gradient method on an affine-quadratic piecewise function. The worst-case function that achieves the first part of Conjecture (4.2.) converges to the affine-quadratic function when κ tends to zero.

Two steps of gradient method. A. Daccache has derived in his master thesis [Dac19] two new conjectures for a tight worst-case bound on the performance criterion $f(x_N) - f(x_*)$, for two steps of gradient method. The first one corresponding to Conjecture (5.1.), is tailored for smooth convex unconstrained minimization. The second one provides an exact worst-case bound for the function accuracy of gradient method applied to smooth strongly convex functions. It corresponds to Conjecture (5.2.). It turns out that both worst-case bounds are expressed as the maximum of four functions depending on the step sizes (h_1, h_2). Each function corresponds to a regime defined by a type of combination between the step sizes: *(S,S)*, *(B,B)*, *(B,S)* and *(S,B)* where S stands for *Small* step sizes and B for *Big* ones. It also turns out that the functions composing the conjecture for smooth strongly convex functions converge to those of the smooth convex case.

Therefore, the second part of this work was to derive functions that achieve the bounds composing Conjectures (5.1.) and (5.2.) when applying two steps of gradient method and for any value of L and R. To proceed, we started by solving (sdp-PEP) for several values of (h_1, h_2) between zero and two. For the case of smooth strongly convex functions, we also added several values of constant μ. We also set the parameters L and R to one, thanks to the homogeneity of the optimal values of (sdp-PEP). Then, we generated our data set with the set of triples $\{(x_i, f_i, g_i)\}_{i\in\{0,1,2,*\}}$ given at each resolution of (sdp-PEP).

Furthermore, we observed that all performance estimation problems that we solved were one-dimensional. Therefore, the functions that achieve those worst-case performances are one-dimensional too. Finally, for each pair (h_1, h_2) and each triple (h_1, h_2, μ), we respectively derived a function $F \in \mathcal{F}_{0,L}(\mathbb{R}^d)$ and a function $F \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ that interpolate the corresponding set of triples. From these interpolating functions, we tried to deduce the worst-case function associated with each regime. We were also inspired by the worst-case functions for one-step of gradient method.

For smooth convex unconstrained minimization, the worst-case functions associated with regimes *(S,S)*, *(B,S)* and *(S,B)* are affine-quadratic piecewise functions. The one corresponding to regime *(B,B)* is purely quadratic and equal to the quadric worst-case function for one step of gradient method. Moreover, the worst-case functions of regimes *(B,S)* and *(S,B)* are not symmetric around the optimal solution. Therefore, they depends on the sign of the initial iterate.

The same conclusions can be drawn for the worst-case functions for smooth strongly convex unconstrained minimization. However, the affine pieces are replaced by quadratic pieces, characterizing the strong convexity. As for the worst-case bounds, we have shown that the worst-case smooth strongly convex functions converge to those of the smooth convex case.

Three steps of gradient method. A. Daccache conjectured in his master thesis [Dac19] that the worst-case bound for three fixed but arbitrary step sizes of gradient method on smooth convex functions, appears to be expressed as the maximum of eight functions, which depend on these step sizes. Each of these functions is maximum for a type of combination between the step sizes (h_1, h_2, h_3). Each step size can be either *Small* (S) or *Big* (B). Therefore, the types of combination are defined by all possibilities between *Small* and *Big* step sizes. Moreover, these types correspond to eight different regimes. We find in [Dac19] seven of the eight functions that seem to compose the worst-case bound. The function that was not identified corresponds to the regime (B, B, S).

The third part of this work was to derive the smooth convex functions that achieve the worst-case bounds associated to the seven regimes, when applying three steps of gradient method and for any value of L and R. We proceeded the same way as for two steps. From the interpolating functions and based on the worst-case functions for two steps, we tried to deduce those seven worst-case functions. It should be noted that the worst-case bound associated to regime (B, B, B) is attained by the same quadratic function as for regime (B, B). The other worst-case functions remain affine-quadratic piecewise and, except for regime (S, S, S), they are all not symmetric around the optimum. Therefore, they depend on the sign of the starting point. We tried to identify the worst-case function associated to the last regime in order to subsequently derive the worst-case bound that was not identified in [Dac19]. Unfortunately, it turns out that the performance estimation problems with step sizes belonging to this regime are two-dimensional. Therefore, we have not been able to compute interpolating functions.

N steps of gradient method. Finally, we can also draw some conclusions on the worst-case bound for N steps of the gradient method with fixed variable step sizes in [Dac19]. First, it seems that the worst-case bound for N steps will be expressed as the maximum of 2^N functions. From this, we also conjectured that the number of worst-case functions is at least 2^N too. In addition to this, we find in [Dac19] the worst-case bounds associated to three of the 2^N regimes. We then derived the corresponding worst-case functions. Identifying all the worst-case functions when N increases is complicated. Indeed, we find that the dimension of the performance estimation problems for some regimes increases with N. Therefore, it becomes very difficult to compute the interpolating functions.

Proofs. All the worst-case functions we have identified rely on the worst-case bounds numerically derived in [Dac19]. The Conjectures (5.1.) and (5.2.) are not proved, therefore our functions are not proved either. However, we succeeded in proving that each worst-case function identified, achieves the corresponding worst-case objective function accuracy when applying the gradient method. Our proofs are generalized for any value of parameters L and R. Once the worst-case bounds on the objective function accuracy for the gradient method are proven, so are our worst-case functions. In the mean time, each of our functions establishes a rigorous lower bound on the performance of the GM with fixed variable step sizes.

8.2 Further research

In this master thesis, we also tried to analyse the performance of a variant of the gradient method. In this variant, the second iterate is computed with the gradient at the initial iterate and at the first iterate. More precisely, iterate x_2 is computed as: $x_2 = x_1 - h_2 \nabla f(x_1) - h_{21} \nabla f(x_0)$. The purpose was to analyze the worst-case performances of this method and compare it with the gradient method with fixed constant step sizes and fixed variable step sizes. Unfortunately, we found with numerical resolutions of (sdp-PEP), that almost half of the solutions are not one-dimensional. Moreover, the first iterate is very often equal to the solution. The second iterate is then further away than the starting point. Therefore, the worst-case performance for such cases is relatively poor. Due to lack of time, we could not go further in the process.

Bibliography

[Dac19] Antoine Daccache. “Performance estimation of the gradient method with fixed arbitrary step sizes”. MA thesis. Université Catholique de Louvain, 2019.

[DT14] Yoel Drori and Marc Teboulle. “Performance of first-order methods for smooth convex minimization: a novel approach”. In: *Mathematical Programming* 145.1 (June 1, 2014), pp. 451–482. ISSN: 1436-4646. DOI: `10.1007/s10107-013-0653-0`. URL: `https://doi.org/10.1007/s10107-013-0653-0` (visited on 06/07/2022).

[KGH21] Ahmed Yassine Kamri, François Glineur, and Julien Hendrickx. “On the minimal and maximal smooth convex interpolating functions”. In: Benelux Workshop on Systems and Control. 2021. URL: `https://dial.uclouvain.be/pr/boreal/object/boreal:258191` (visited on 07/19/2022).

[Löf04] J. Löfberg. “YALMIP : A Toolbox for Modeling and Optimization in MATLAB”. In: In Proceedings of the CACSD Conference. 2004.

[Mos10] APS Mosek. “The MOSEK optimization software”. In: *Online at http://www.mosek.com* 54 (2010).

[Nes18] Yurii Nesterov. *Lectures on Convex Optimization*. Vol. 137. Springer Optimization and Its Applications. Cham: Springer International Publishing, 2018. ISBN: 978-3-319-91577-7 978-3-319-91578-4. DOI: `10.1007/978-3-319-91578-4`. URL: `http://link.springer.com/10.1007/978-3-319-91578-4` (visited on 06/07/2022).

[Stu99] Jos F. Sturm. “Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones”. In: *Optimization methods and software* 11.1 (1999), pp. 625–653.

[Tay17] Adrien Taylor. “Convex interpolation and performance estimation of first-order methods for convex optimization”. PhD thesis. UCL - Université Catholique de Louvain, 2017. URL: `https://dial.uclouvain.be/pr/boreal/object/boreal:182881` (visited on 06/07/2022).

[THG17a] Adrien Taylor, Julien Hendrickx, and François Glineur. “Performance Estimation Toolbox (PESTO): automated worst-case analysis of first-order optimization methods”. In: Proceedings of the 56th IEEE Conference on Decision and Control (CDC 2017). 2017. URL: `https://dial.uclouvain.be/pr/boreal/fr/object/boreal%3A192237` (visited on 06/07/2022).

[THG17b] Adrien B. Taylor, Julien M. Hendrickx, and François Glineur. “Exact Worst-Case Performance of First-Order Methods for Composite Convex Optimization”. In: *SIAM Journal on Optimization* 27.3 (Jan. 2017). Publisher: Society for Industrial and Applied Mathematics, pp. 1283–1313. ISSN: 1052-6234. DOI: `10.1137/16M108104X`. URL: `https://epubs.siam.org/doi/10.1137/16M108104X` (visited on 08/19/2022).

[THG17c] Adrien B. Taylor, Julien M. Hendrickx, and François Glineur. "Smooth strongly convex interpolation and exact worst-case performance of first-order methods". In: *Mathematical Programming* 161.1 (Jan. 1, 2017), pp. 307–345. ISSN: 1436-4646. DOI: 10.1007/s10107-016-1009-3. URL: https://doi.org/10.1007/s10107-016-1009-3 (visited on 06/07/2022).

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**