# Worst-case analysis of efficient first-order methods

Mathieu Barré

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'École Normale Supérieure

# Worst-Case Analysis of Efficient First-Order Methods


Soutenue par
**Mathieu Barré**
Le 6 octobre 2021


École doctorale n°386
**École Doctorale Sciences Mathématiques de Paris Centre**

Spécialité
**Mathématiques Appliquées**

Composition du jury :

| | |
|---|---|
| Francis Bach<br>Directeur de Recherche, INRIA | *Président du jury* |
| François Glineur<br>Professeur, UCLouvain | *Rapporteur* |
| Silvia Villa<br>Professeure,<br>Université de Gênes | *Rapportrice* |
| Alexandre d'Aspremont<br>Directeur de Recherche, CNRS | *Directeur de thèse* |

1794
ENS | PSL

# Resumé

De nombreuses applications modernes reposent sur la résolution de problèmes d'optimisations (par exemple, en biologie numérique, en mécanique, en finance), faisant des méthodes d'optimisation des outils essentiels dans de nombreux domaines scientifiques. Apporter des garanties sur le comportement de ces méthodes constitue donc un axe de recherche important.

Une façon classique d'analyser un algorithme d'optimisation consiste à étudier son comportement dans le pire cas. C'est à dire, donner des garanties sur son comportement (par exemple sa vitesse de convergence) qui soient indépendantes de la fonction en entrée de l'algorithme et vraies pour toutes les fonctions dans une classe donnée. Cette thèse se concentre sur l'analyse en pire cas de quelques méthodes du premier ordre réputées pour leur efficacité.

Nous commençons par étudier les méthodes d'accélération d'Anderson, pour lesquelles nous donnons de nouvelles bornes de pire cas qui permettent de garantir précisément et explicitement quand l'accélération a lieu. Pour obtenir ces garanties, nous fournissons des majorations sur une variation du problème d'optimisation polynomiale de Tchebychev, dont nous pensons qu'elles constituent un résultat indépendant.

Ensuite, nous prolongeons l'étude des Problèmes d'Estimation de Performances (PEP), développés à l'origine pour analyser les algorithmes d'optimisation à pas fixes, à l'analyse des méthodes adaptatives. En particulier, nous illustrons ces développements à travers l'étude des comportements en pire cas de la descente de gradient avec pas de Polyak, qui utilise la norme des gradients et les valeurs prises par la fonction objectif, ainsi que d'une nouvelle version accélérée. Nous détaillons aussi cette approche sur d'autres algorithmes adaptatifs standards.

Enfin, la dernière contribution de cette thèse est de développer plus avant la méthodologie PEP pour l'analyse des méthodes du premier ordre se basant sur des opérations proximales inexactes. En utilisant cette approche, nous définissons des algorithmes dont les garanties en pire cas ont été optimisées et nous fournissons des analyses de pire cas pour quelques méthodes présentes dans la littérature.

# Abstract

Many modern applications rely on solving optimization problems (e.g., computational biology, mechanics, finance), establishing optimization methods as crucial tools in many scientific fields. Providing guarantees on the (hopefully good) behaviors of these methods is therefore of significant interest.

A standard way of analyzing optimization algorithms consists in worst-case reasoning. That is, providing guarantees on the behavior of an algorithm (e.g. its convergence speed), that are independent of the function on which the algorithm is applied and true for every function in a particular class. This thesis aims at providing worst-case analyses of a few efficient first-order optimization methods.

We start by the study of Anderson acceleration methods, for which we provide new explicit worst-case bounds guaranteeing precisely when acceleration occurs. We obtained these guarantees by providing upper bounds on a variation of the classical Chebyshev optimization problem on polynomials, that we believe of independent interest.

Then, we extend the Performance Estimation Problem (PEP) framework, that was originally designed for principled analyses of fixed-step algorithms, to study first-order methods with adaptive parameters. This is illustrated in particular through the worst-case analyses of the canonical gradient method with Polyak step sizes that use gradient norms and function values information, and of an accelerated version of it. The approach is also presented on other standard adaptive algorithms.

Finally, the last contribution of this thesis is to further develop the PEP methodology for analyzing first-order methods relying on inexact proximal computations. Using this framework, we produce algorithms with optimized worst-case guarantees and provide (numerical and analytical) worst-case bounds for some standard algorithms in the literature.

# Remerciements

Je tiens tout d'abord à remercier mon directeur de thèse Alexandre d'Aspremont pour sa confiance et son soutien tout au long de cette thèse. Il a su plusieurs fois me remotiver et contrebalancer mes avis parfois très tranchés sur mon travail, et pour cela je lui suis très reconnaissant.

Je remercie également François Glineur et Silvia Villa d'avoir accepté de rapporter cette thèse. Leurs retours détaillés et leurs multiples questions m'ont permis de gagner en recul sur le sujet.

Je souhaite aussi remercier Francis Bach, d'abord pour avoir présidé le jury, mais aussi et surtout pour avoir été un excellent chef d'équipe au sein de l'Inria. Il a toujours été attentif au bien-être des membres de l'équipe, et je le tiens en grande partie pour responsable de la très bonne ambiance régnant au 4e étage.

Je tiens particulièrement à remercier Adrien Taylor avec qui ce fut un plaisir de travailler et de "cohabiter" à l'Inria. Il m'a initié à la magie des PEPs, et plus généralement m'a énormément appris en optimisation, grâce à sa pédagogie, sa gentillesse et sa patience infinies. Nos discussions ne se résumaient pas qu'au travail et c'est grâce à lui par exemple que je lève maintenant la tête quand j'entends un oiseau chanter !

Un grand merci aux nombreux occupants du bureau C407 que j'ai eu le plaisir de côtoyer. Ce fut une joie de démarrer l'aventure aux côtés de Damien Scieur et Antoine Recanati. Merci à Thomas Kerdreux, Grégoire Mialon et Radu Dragomir pour avoir allié ambiance studieuse, discussions animées et lecture d'avenir dans la corbeille à papier pendant presque 3 ans. Enfin merci à Manon Romain avec qui ce fut très agréable d'échanger.

Un merci particulier à mes collègues de Halo, Thomas Eboli et Rémi Jézéquel. Je remercie aussi plus généralement les membres du 4e étage avec qui la vie de tous les jours au bureau était gaie et amusante, Alex N, Antoine B, Ben D, Bruno L, Céline M, Hadrien H, Héloïse B, Loucas P, Oumayma B, Raphaël B, Robin S, Ulysse M, Yana H, Yann L et bien d'autres.

Merci à mes compagnons de galère Jules, Mickaël, Nidham et Quentin pour tous nos débats et discussions souvent portés autour de la thèse, et merci à Matthieu et Mélanie de nous avoir supportés. Je remercie aussi coach Nicolas pour m'avoir maintenu en forme tout ce temps.

Enfin, merci à ma famille pour le soutien continuel qu'ils m'ont apporté, merci en particulier à mes parents et ma grand-mère d'avoir fait le déplacement pour ce moment si particulier qu'est la soutenance.

Pour conclure, merci à Marion avec qui je partage ma vie depuis plus de 3 ans et qui a vécu avec moi cette étape de ma vie. Merci de m'avoir apaisé et soutenu pendant toute cette période qui a été parfois difficile pour toi aussi. J'espère pouvoir te rendre la pareille dans les 55 prochaines années (au moins !) que nous allons vivre ensemble.

# Notations

$\mathbb{N}$ is the set of integers.

$\mathbb{R}$ is the set of real numbers.

$\mathbb{R}^d$ is the set of real vectors of dimension $d$.

$\mathbb{R}_k[X]$ is the set of polynomials of degree smaller than $k \in \mathbb{N}$ with real coefficients.

$\mathbb{R}^{d_1 \times d_2}$ is the set of real matrices of size $d_1$ by $d_2$.

$\mathbf{S}_d$ is the set of symmetric matrices of size $d$ by $d$.

$\mathbf{S}_d^+$ is the set of positive semidefinite symmetric matrices of size $d$ by $d$.

$\mathrm{Tr}(A)$ is the trace of matrix $A$.

$A^T$ is the transpose of matrix $A$.

$I$ is the identity matrix or identity application depending on the context.

1 is a vector of ones with size depending on the context.

$\|\cdot\|$ is the Euclidean norm on vectors and the spectral norm on linear operators.

$\|\cdot\|_1$ is the $\ell_1$-norm on vectors. When applied to polynomials, it is the $\ell_1$-norm of the vector of coefficients.

$\|\cdot\|_\infty$ is the $\ell_\infty$-norm on vectors.

$\|\cdot\|_F$ is the Frobenius norm on matrices.

$\mathrm{sign}(x)$ is the sign of $x \in \mathbb{R}$, and $\mathrm{sign}(0) = 0$.

$\mathcal{F}(X, Y)$ is the set of functions defined on $X$ with values in $Y$.

$\mathcal{F}_{0,\infty}(\mathbb{R}^d)$ is the set of closed convex and proper functions over $\mathbb{R}^d$ (see Definition 3.1.1).

$\partial f$ is the subgradient mapping of $f$ (see Definition 3.1.2).

$\mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$ for $0 \leq \mu \leq L \leq +\infty$ is the set of closed $\mu$-strongly convex and proper functions over $\mathbb{R}^d$ with $L$-Lipschitz subgradient mapping (see Definition 3.1.15).

$\mathbf{dom}\, f$ is the domain of $f$, i.e. $\{x,\ f(x) < +\infty\}$.

$i_C$ is the indicator function of a set $C$, i.e. $i_C(x) = 0$ when $x \in C$ and $+\infty$ otherwise.

$f^*$ is the Fenchel conjugate of the function $f$ (see Definition 3.1.12).

$\mathrm{ri}(C)$ is the relative interior of the set $C \subset \mathbb{R}^d$.

# Contents

# Chapter 1

# Introduction

## 1.1 Introduction to optimization

Optimality is a central concern in many scientific fields. Indeed, we model Nature using optimality properties (e.g., Fermat's principle in optic, least action principle in mechanics or geometry optimization in chemistry) and applied these models in human made fields as economy or finance. Describing formally what is optimal and being able to reach optimality constitute the essence of mathematical optimization. Many modern problems can be formulated as optimization problems and solved using optimization algorithms, making it a crucial tool in modern science.

Mathematically, an optimization problem can be formalized as

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{such that} & x \in C, \end{array} \tag{1.1}$$

where $f : X \to \mathbb{R} \cup \{-\infty, +\infty\}$ is an input objective function defined on some space $X$ and $C \subset X$ is a constrained set. In particular, when it admits a solution, optimization aims at solving this problem. For instance, in molecular geometry optimization, the objective function corresponds to the energy of the molecule, the variable $x$ represents the relative positions of the atoms that compose it, and $C$ some physical constraints on the positions. In that context, being able to find a solution to (1.1) would allow predicting the atomic structure of a particular molecule.

In general, the cost (or difficulty) of solving an optimization problem as (1.1) depends on the particular structures of $f$ and $C$. Developing methods to tackle efficiently this problem under its general form is out of reach. A large amount of work in optimization has been dedicated to cases where $f$ and $C$ are convex, already modeling a large body of problems (e.g. Boyd and Vandenberghe [2004], Palomar and Eldar [2010], Chambolle and Pock [2016]). Moreover, convex optimization constitutes a building block in the resolution of more complex problems. Convexity definitions are recalled here and more precisely in Chapter 3.

**Definition 1.1.1** (Convex set)**.** *Let $X$ be a real vector space, and $C \subset X$. $X$ is convex if and only if*

$$\forall t \in [0,1],\ \forall x, y \in C,\ tx + (1-t)y \in C.$$

Convexity of real valued functions can be obtained from this definition as follows.

**Definition 1.1.2** (Convex function)**.** *Let $X$ be a real vector space and $f : X \to \mathbb{R} \cup \{-\infty, +\infty\}$, the epigraph of $f$ is defined as*

$$\operatorname{epi}(f) = \{(x, \alpha) \in X \times \mathbb{R} \text{ s.t. } f(x) \leq \alpha\},$$

*and $f$ is a convex function if and only if* $\mathrm{epi}(f)$ *is a convex set.*

An important effect of convexity is to make local discussions on optimality global (see e.g. convex optimization textbooks Polyak [1987], Nemirovsky [1995], Nesterov [2018] or Nocedal and Wright [2006] for more practical discussions). In this manuscript, we focus on convex optimization in finite dimension, i.e. objective functions and constrained sets will be convex and the ambient space $X$ will be some real vector space $\mathbb{R}^d$. Additional assumptions as closeness (i.e. closed level sets) or being proper (i.e. function is not equal to $+\infty$) are often used in order to avoid pathological cases (we refer to Chapter 3 for precise definitions).

In the following, we give a high level overview on the class of optimization methods that are studied in this thesis.

### 1.1.1 First-order methods

When the objective function of an optimization problem is regular enough, first-order information (i.e. local linear approximation of $f$) can be of great help. Indeed, the gradient (or a subgradient for convex functions [Rockafellar, 1996]) at some point $x \in \mathbb{R}^d$ provides a direction in which the objective function is locally increasing (and locally decreasing in the opposite direction). The idea of methods that start from an initial guess $x_0 \in \mathbb{R}^d$ and update iteratively the current iterate by following first-order directions dates back to the 18th century with the work of Cauchy [Cauchy, 1847] (see e.g. Lemaréchal [2012] for a short historical survey). They are usually referred to as first-order (or gradient) methods. For instance, given a differentiable function, gradient descent builds a sequence of iterates $\{x_k\}_k$ using updates of the form

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k), \text{ for } k \geq 0,$$

with $x_0 \in \mathbb{R}^d$ some starting point and $\{\alpha_k\}_k$ a sequence of step sizes (often chosen constant or decreasing). Under some condition on $f$ and on $\{\alpha_k\}_k$, the sequence of iterates $\{x_k\}_k$ converges toward some stationary point (local minima) of $f$. We refer to the recent textbook from Beck [2017] for developments on many first-order methods.

First-order methods generally require more iterations than algorithms relying on second-order information to reach the same level of accuracy but iterations are often much cheaper. This makes these algorithms particularly appealing for high dimensional applications with moderate target accuracy (e.g. for inverse problems or machine learning).

Finally, these optimization methods typically assess input functions through their evaluations at particular points or evaluations of their derivatives, being agnostic about the exact structure or properties of the entry. These algorithms benefit of being generic, in the sense that they can be applied to a wide range of problems without particular tuning, but of course can be less efficient than some specific algorithms on a particular application. For further information and a detailed treatment on "black-box" optimization, we refer to the classical textbooks Nemirovsky [1995], Nesterov [2018].

### 1.1.2 Worst-case guarantees of optimization methods

Worst-case reasoning is a standard way of analyzing optimization algorithms. Such analyses aim at providing guarantees on the behavior of an algorithm (e.g. its convergence speed), that are independent of the function on which the algorithm is applied and true for every functions in a particular class. Although worst-case analyses may not always reflect what can actually be observed when instantiating an optimization method on a particular entry (as not all the entries may produce the worst possible behavior), they constitute strong theoretical insurances on the well-functioning of a method for a range

of possible inputs. Furthermore, worst-case guarantees that are independent of the input functions can usually be evaluated in advance without a precise knowledge of the function structure. This is particularly useful in practice for estimating e.g. worst-case running time of a method.

Let $\mathcal{F}$ be a subset of $\mathcal{F}(\mathbb{R}^d, \mathbb{R} \cup \{-\infty, +\infty\})$ such that (1.1) admits a solution when $f \in \mathcal{F}$, and consider an optimization method $\mathcal{M}$ that takes as entries a starting point $x_0 \in \mathbb{R}^d$, a function $f \in \mathcal{F}$, an iteration number $N \in \mathbb{N}^*$ and output an iterate $x_N \in \mathbb{R}^d$. That is, $\mathcal{M} : \mathbb{R}^d \times \mathcal{F} \times \mathbb{N}^* \to \mathbb{R}^d$. Worst-case guarantees for the method $\mathcal{M}$ on the class of function $\mathcal{F}$ typically look like

$$\Phi_{\text{obj}}(x_N, f) \leq \tau(N, \mathcal{F})\Phi_{\text{init}}(x_0, f), \text{ for all } f \in \mathcal{F}, \, x_0 \in \mathbb{R}^d, \tag{1.2}$$

where $x_N = \mathcal{M}(x_0, f, N)$, $\Phi_{\text{obj}}$ an objective criterion that quantify how far is $x_N$ from being a solution to (1.1), $\Phi_{\text{init}}$ an initial condition quantifying how far is $x_0$ from solving (1.1) and $\tau$ a worst-case bound (hopefully decreasing toward 0 with $N$). For instance, when $\mathcal{F}$ is the class of differentiable convex functions over $\mathbb{R}^d$ with $L$-Lipschitz gradient, and $\mathcal{M}$ is a gradient descent method with well chosen step size, worst-case guarantees take the form (see Drori and Teboulle [2014, Theorem 3.1])

$$f(x_N) - f(x_*) \leq \tfrac{L}{4N+2}\|x_0 - x_*\|^2, \text{ for all } f \in \mathcal{F}, \, x_0 \in \mathbb{R}^d,$$

with $x_* \in \operatorname{argmin}_x f(x)$, ensuring that the objective value at iterate $x_N$ converge to the optimal one at a rate at most as large as $O(N^{-1})$.

The bound $\tau(N, \mathcal{F})$ should control the behavior of the method over all possible inputs in a particular class and therefore, this corresponds to study the worst possible scenario for the method (i.e. worst objective function and worst initial iterate in this context). These worst-case instances can be described by the following optimization problem,

$$\max_{\substack{x_0 \in \mathbb{R}^d \\ f \in \mathcal{F} \\ \Phi_{\text{init}}(x_0, f) \neq 0}} \frac{\Phi_{\text{obj}}(\mathcal{M}(x_0, f, N), f)}{\Phi_{\text{init}}(x_0, f)}, \tag{1.3}$$

and the optimal value of (1.3) corresponds to the smallest $\tau(N, \mathcal{F})$ that satisfies (1.2).

In that case, $\tau$ is dependent on the class of function $\mathcal{F}$, and therefore on the ambient dimension $d$. However, we are often interested in worst-case guarantees valid for a set of function classes $\{\mathcal{F}_d\}_{d \in \mathbb{N}^*}$ with $\mathcal{F}_d \subset \mathcal{F}(\mathbb{R}^d, \mathbb{R})$ leading to bounds $\tau$ that no longer depends on $d$.

In the next section, we introduce a recent framework to study worst-case behaviors of many optimization methods in a systematic way.

### 1.1.3 Computer aided worst-case analysis

Worst-case behavior of an optimization method is sometimes hard to intuit as worst-case instances may be pathological and might not reflect observations on standard inputs. In addition, optimization methods with sharp worst-case bounds tend to require a high-level of technicality in their analysis or their design, that may be difficult to complete by hand.

Computer aided worst-case analysis aims at helping optimizers to study the behavior of first-order optimization methods. In particular, the *performance estimation problem* (PEP) framework initiated by Drori and Teboulle [2014] transcribes the worst-case guarantees of a given fixed-step first-order method into an optimization problem (similar to 1.3) that is reformulated as a semidefinite program and solved numerically. These techniques were used, among others, to provide new analyses of gradient and fast gradient methods [Nesterov, 1983].

This methodology was extended with the work of Taylor et al. [2017c], Taylor [2017]. Functional interpolation results are introduced to guarantee tightness of the semidefinite reformulation, that is, ensuring that the numerical solutions are indeed the smallest worst-case bounds. This framework allows performing worst-case analysis of first-order methods on smooth and strongly convex function in a principled way. Feasible points to the primal semidefinite problem correspond to matching examples (i.e., worst-case instances: functions and iterates). In addition, it can be shown that feasible points to the dual problem can be used to write proofs of worst-case guarantees (i.e., show how to combine inequalities satisfied by the input functions).

The possibility of solving those problems numerically essentially allows sampling worst-case examples and proofs for given problem parameters (for example step sizes, smoothness or strong convexity levels).

The PEP framework has been used to design several optimization methods with optimized guarantees as the Optimized Gradient Method (OGM) from Kim and Fessler [2016] for smooth convex optimization, the strongly convex version ITEM from Taylor and Drori [2021] as well as other algorithms [Drori and Teboulle, 2016, Kim and Fessler, 2018, Drori and Taylor, 2020, Kim and Fessler, 2021, Kim, 2021, Ryu and Vũ, 2019, Lee et al., 2021].

The PEP approach has been extended in many directions by e.g. incorporating Lyapunov (or potential functions) arguments [Taylor et al., 2018b, Taylor and Bach, 2019], analyzing methods based on inexact gradient computations [Klerk et al., 2017, 2020], Bregman gradient methods [Dragomir et al., 2021] or by studying worst-case guarantees for monotone inclusion problems such as splitting methods [Ryu et al., 2020, Ryu and Vũ, 2019], proximal point algorithms [Gu and Yang, 2019a,b, Kim, 2021] or fixed point iterations [Lieder, 2020].

Note that the PEP approach is closely related with *Integral Quadratic Constraints* (IQC) originating from control theory [Megretski and Rantzer, 1997] and further developed to analyze and design optimization algorithms [Lessard et al., 2016, Hu and Lessard, 2017, Cyrus et al., 2018, Van Scoy et al., 2018].

The PEP methodology is a central tool in this manuscript, especially in Chapters 4 and 5 where it is extended to different contexts (adaptive methods in Chapter 4 and inexact proximal computations in Chapter 5). A short introduction to the main results of performance estimation can be found in Chapter 3 for readers who are not familiar with this approach.

## 1.2 Adaptive first-order methods

In this section, we focus on the particular type of first-order optimization methods that can be considered as "adaptive". As opposed to fixed-step methods such as standard gradient descent with predefined step sizes or Nesterov's accelerated methods [Nesterov, 2018] that combine first-order information regardless of the methods inputs, adaptive algorithms aim at adjusting their behavior according to the current state of the method. Adaptive strategies often exhibit much better practical performances than their nonadaptive counter part, although it is not always the case for theoretical guarantees.

### 1.2.1 A short overview on first-order adaptive methods

Many adaptive methods estimate some regularity parameters of the function being minimized, which can vary depending on the region in which the iterates evolve. Indeed, regularity of the function might be different when we are close or far from the optimum.

In order to improve the accelerated methods performances, line searches strategies are often used to estimate local Lipschitz constant of the gradient [Nesterov, 1983, Beck and Teboulle, 2009], leading to significant improvement in practical performances.

However, some regularity properties such as strong convexity or related quantities (e.g. Hölderian error bounds [Bolte et al., 2007]) are required by accelerated methods to reach optimal complexity bounds Nesterov [2018], Arjevani and Shamir [2016] and cannot be estimated using simple line searches. Poor approximation of these parameters often lead to deteriorated convergence speeds. An important line of work has been dedicated to the study of restarts strategies. They consist in plugging back the output of an fixed-step accelerated method as new input for the same algorithm. Initiated by the work of Nemirovskii and Nesterov [1985], Nesterov [2013], O'Donoghue and Candes [2015], several restarts schedules have been proposed in e.g. Lin and Xiao [2014], Fercoq and Qu [2016], Roulet and d'Aspremont [2020], Renegar and Grimmer [2021] together with theoretical guarantees on their adaptability. These methods are not totally satisfying as they often rely on extra parameters that have to be tuned.

Another important class of adaptive methods includes so called quasi-Newton methods. These methods intend to mimic the behavior of Newton's method without accessing full second-order information, by constructing estimates of the Hessian at the current iterates from variations in the past gradients. These methods are more costly than simple gradient or accelerated gradient methods but exhibit impressive results in many applications. L-BFGS [Liu and Nocedal, 1989] is probably the most celebrated quasi-Newton method and is a building block of many optimization solvers, although only local theoretical improvements compared to simple gradient scheme have been provided yet. We refer to Dennis and Moré [1977], Dennis and Schnabel [1996] for detailed treatments on quasi-Newton methods.

Among first-order adaptive methods, the conjugate gradient method [Hestenes and Stiefel, 1952] has drawn the interest of the optimization community for many decades. Originally designed for the minimization of convex quadratic functions (i.e. functions with affine gradient mappings) for which it exhibits optimal convergence guarantees, it has been extended to the nonquadratic setting with e.g. Fletcher and Reeves [1964], Polyak [1969], Fletcher [1987], Dai and Yuan [1999]. These methods only use first-order information and rely on exact or approximate line searches. Although they demonstrate fast empirical convergence, few theoretical results have been obtained on the convergence rate of these methods when applied to nonquadratic functions. We refer to the nice survey by Hager and Zhang [2006] for a comprehensive review.

Furthermore, several gradient methods with simple adaptive step sizes as those of Polyak [1987, Section 5.3.2], Barzilai and Borwein [1988] or Malitsky and Mishchenko [2019] have been proposed to adapt to the local geometry of the problem.

Finally, in the context of stochastic optimization (that is not considered in this manuscript), several adaptive gradient methods have been developed with e.g. the celebrated *AdaGrad* [Duchi et al., 2011] or *Adam* [Kingma and Ba, 2014] methods.

In the following, we focus on two particular adaptive methods that are Anderson acceleration schemes and gradient method with Polyak step sizes.

### 1.2.2 Anderson acceleration

Extrapolation techniques [Anderson, 1965, Pulay, 1980, Sidi et al., 1986] are popular tools for speeding up convergence of iterative processes (e.g. first-order methods) towards their limit points. Anderson acceleration proceeds by extrapolating a better approximation of the limit using a weighted combinations of previous iterates. These weights are obtained as solutions of a simple quadratic program also

depending on previous iterates.

In particular, given a converging iterative fixed point process $F : \mathbb{R}^d \to \mathbb{R}^d$ producing iterates $\{x_i\}_{i=0,\dots,k}$ such that

$$x_{i+1} = F(x_i), \text{ for } i = 0, \dots, k-1,$$

Anderson acceleration produces an extrapolated point

$$x_e = \sum_{i=0}^{k-1} c_i^* x_i,$$

where the $c_i^*$'s are solution to

$$c^* = \underset{\substack{c \in \mathbb{R}^k \\ 1^T c = 1}}{\operatorname{argmin}} \|\sum_{i=0}^{k-1} c_i (x_{i+1} - x_i)\|, \tag{1.4}$$

which can be reformulated as a quadratic problem.

Anderson acceleration was originally developed for accelerating convergence of iterative methods for solving linear systems which corresponds to an affine mapping $F$. In that case, (1.4) can be written as

$$c^* = \underset{\substack{c \in \mathbb{R}^k \\ 1^T c = 1}}{\operatorname{argmin}} \|(F - I) \sum_{i=0}^{k-1} c_i x_i\|, \tag{1.5}$$

and $x_e$ corresponds to the combination with the smallest residual. For instance, when $F$ is a gradient step operator of a quadratic function $f$ that is $F(x) = x - \alpha \nabla f(x)$, (1.5) aims at finding the combination with smallest gradient norm. Theory is the case of affine $F$ is well furnished but becomes fuzzier in the presence of nonlinearity. For nonlinear $F$, problem (1.4) can suffer from really poor conditioning and convergence of the extrapolation scheme can break.

Many corrections have been proposed in the literature to stabilize these extrapolation processes when dealing with nonlinear mappings. For instance, imposing some linear independence assumptions on the iterates $\{x_k\}_k$ allows deriving convergence guarantees [Rohwedder and Schneider, 2011, Brezinski and Redivo-Zaglia, 2019, Pollock and Rebholz, 2019]. Assumptions on the boundedness of the extrapolation weights have also been considered in order to provide analyses in the nonlinear setting [Toth and Kelley, 2015, Ouyang et al., 2020]. These strategies require assumptions to hold that are often impossible to check without actually running the method. A regularization of problem (1.4) has been proposed by Scieur et al. [2016, 2020] in order to stabilize the method, together with asymptotic convergence guarantees. In this work, the authors mainly focus on the acceleration of gradient descent procedure, and many extensions have followed with e.g. [Scieur et al., 2017, Mai and Johansson, 2020, Bollapragada et al., 2019, Poon and Liang, 2019, Bertrand and Massias, 2021].

In Chapter 2, we provide an Anderson acceleration scheme that modifies problem (1.4) in the spirit of Scieur et al. [2020], by imposing hard bounds on the extrapolation weights. Using these constraints we can obtain explicit local linear convergence rates for the method in a nonlinear setting.

### 1.2.3 Polyak step sizes

Gradient descent with Polyak step sizes [Polyak, 1987, Section 5.3.2] is a simple adaptive method that consists in gradient updates with a step size proportional to the difference between the current objective value and its minimal value. It was first introduced in the context of subgradient methods

[Polyak, 1987, Nedic and Bertsekas, 2001, Boyd et al., 2003], where efficient step sizes policies are sometimes difficult to determine.

Given a differentiable convex function $f$, gradient updates with Polyak step sizes can be written as

$$x_{k+1} = x_k - \frac{f(x_k)-f_*}{\|\nabla f(x_k)\|^2}\nabla f(x_k),$$

where $f_* = \min_x f(x)$ has to be known in advance. This choice of step size naturally follows from the convexity assumption on $f$. Indeed, when performing a gradient step of the form

$$x_{k+1} = x_k - \alpha\nabla f(x_k),$$

with $\alpha \in \mathbb{R}$, the distance between the new iterates $x_{k+1}$ and $x_* \in \operatorname{argmin}_x f(x)$ can be expressed as

$$\|x_{k+1} - x_*\|^2 = \|x_k - x_*\|^2 - 2\alpha\langle\nabla f(x_k), x_k - x_*\rangle + \alpha^2\|\nabla f(x_k)\|^2.$$

Convexity of $f$ guarantees that $f(x_k) + \langle\nabla f(x_k), x_* - x_k\rangle \leq f_*$ which implies that

$$\|x_{k+1} - x_*\|^2 \leq \|x_k - x_*\|^2 - 2\alpha(f(x_k) - f_*) + \alpha^2\|\nabla f(x_k)\|^2,$$

and minimizing the right-hand side in $\alpha$ leads to an optimal step size $\alpha = (f(x_k) - f_*)/\|\nabla f(x_k)\|^2$. Theoretical guarantees for gradient method with Polyak step size are similar to its nonadaptive counterparts (see e.g. Hazan and Kakade [2019]). However, when it can be applied, it often exhibits much better empirical performances.

Polyak step sizes have also recently witnessed regained interest in the context of stochastic optimization where knowledge of $f_*$ can be made less restrictive [Loizou et al., 2021, Gower et al., 2021].

This simple step sizes policy is used as a canonical example in Chapter 4 to develop the performance estimation methodology for adaptive methods.

## 1.3 Proximal methods and inexactness

Proximal operations were originally introduced in optimization by Martinet [1970, 1972] and Rockafellar [1976a,b], and constitute base primitives in many optimization methods. These operations were originally defined in the context of vector valued mappings (see e.g. Ryu and Boyd [2016] for an excellent survey), but we focus on their optimization versions. Given a closed convex and proper function $f$ over $\mathbb{R}^d$, the proximal operator with step size $\lambda \geq 0$ associated to $f$ corresponds to the mapping

$$\operatorname{prox}_{\lambda f}(z) = \operatorname*{argmin}_{x\in\mathbb{R}^d} \lambda f(x) + \tfrac{1}{2}\|x - z\|^2, \tag{Prox}$$

which is well defined for all $z \in \mathbb{R}^d$ using the assumptions on $f$. Therefore, proximal operations in that case, consist in solving a strongly convex minimization problem (i.e. convex + squared euclidean norm).

### 1.3.1 A short overview of proximal methods

Proximal operations are particularly useful for algorithms dedicated to the minimization of sum of functions and/or when the objective functions are nondifferentiable. It is a crucial tool in e.g. proximal-gradient methods [Bruck Jr, 1975, Lions and Mercier, 1979, Passty, 1979, Nesterov, 2013], Douglas-Rachford splitting [Douglas and Rachford, 1956, Lions and Mercier, 1979, Eckstein and Bertsekas,

1992] or the alternating direction method of multipliers [Fortin and Glowinski, 1983, Gabay, 1983], as well as in many other splitting methods [Eckstein, 1989, Combettes and Pesquet, 2011, Davis and Yin, 2017]. Proximal operations are also central in the development of augmented Lagrangian methods [Rockafellar, 1973, 1976a, Iusem, 1999].

More recently, these proximal operations have been used to accelerate slow algorithms in a systematic way (see e.g the *Catalyst* framework from Lin et al. [2015, 2018] or the monograph Aspremont et al. [2021, Section 5]), or to take advantage of high-order information with e.g. second-order [Monteiro and Svaiter, 2013, Section 6] or higher-order tensor methods [Nesterov, 2019].

For more developments on proximal algorithms, we refer to the recent surveys Combettes and Pesquet [2011], Parikh and Boyd [2014], Ryu and Boyd [2016].

### 1.3.2 Inexact proximal computations

In many applications, the proximal operations (Prox) involved in optimization methods have closed form expressions. This is for instance the case of the popular *Lasso* problem in statistics [Tibshirani, 1996], where the proximal mapping of the $\ell_1$-norm has an exact formula (see e.g., Chierchia et al. [2020] for a comprehensive list of formulae for various proximal mappings). However, this is far from being a generic case, and in most situations, proximal operations have to be approximated by e.g. using an optimization method to solve (Prox).

We wish to perform inexact proximal computations (i.e. solving (Prox)) in a clever way. That is, not solving it to an unnecessary precision. In particular, when inexact proximal operations are used inside an optimization method, it appears natural to start solving the inner proximal problems with low precision and increase it as we get closer to the solution to the global problem. Choosing the accuracy with which each inner proximal problem should be solved can be a challenging task.

Inexact proximal operations have been early considered with Rockafellar [1976a,b]. It has been tackled in many works since, with e.g. inexact proximal point algorithm [Burachik et al., 1997, Eckstein, 1998] and its accelerated variants [Güler, 1992, Monteiro and Svaiter, 2013, Salzo and Villa, 2012], (accelerated) proximal gradient methods [Schmidt et al., 2011, Villa et al., 2013, Millán and Machado, 2019], Douglas-Rachford splitting [Eckstein and Yao, 2018, Svaiter, 2018, Alves et al., 2019], online optimization [Dixit et al., 2019, Ajalloeian et al., 2020] or meta algorithms as Lin et al. [2015, 2018] that take advantage of inexact proximal computations to accelerate optimization methods.

### 1.3.3 Example: a simple inexact proximal method

For illustrative purposes, we present a basic example of inexact proximal operation. Given a closed convex and proper function $f$ over $\mathbb{R}^d$, and an initial point $z \in \mathbb{R}^d$, let $x \in \mathbb{R}^d$ be the proximal step such that $x = \operatorname{prox}_{\lambda f}(z)$ with $\lambda$ a nonnegative step size. Using the notions of subgradients for closed convex and proper functions Rockafellar [1996, §23], this proximal step can be written explicitly as

$$x = z - \lambda s_x,$$

with $s_x$ a subgradient of $f$ at $x$ (when $f$ is differentiable $s_x = \nabla f(x)$). We can define an approximate proximal step $x \approx \operatorname{prox}_{\lambda f}(x)$ by considering a point $x \in \mathbb{R}^d$ that satisfies

$$x = z - \lambda(s_x + e),$$

with $e \in \mathbb{R}^d$ some error term. The quality of this approximation can be monitored by controlling e.g. $\|e\|$. When this approximate proximal step is used inside an optimization method, setting the

inexactness level (i.e. a bound on $\|e\|$ in that case) is a crucial task as it largely influence practical performances (see e.g. Rockafellar [1976a, criteria (A') and (B')] for different strategies).

We review different measures of inexactness for proximal computations in Chapter 5. In addition, we provide a principled way of analyzing optimization methods relying on inexact proximal operations, based on the performance estimation methodology [Drori and Teboulle, 2014, Taylor et al., 2017c]. We also present and analyze some inexact proximal methods that have been obtained through this framework in Chapter 6.

## 1.4 Thesis outline and contributions

In this last section, we describe the organization of the manuscript, together with scientific contributions. Each chapter is supposed to be self-sufficient but we recommend reading Chapter 3 before Chapter 4 or Chapter 5. Although Chapters 5 and 6 can be considered separately, we prescribe reading Chapter 5 before Chapter 6.

### Chapters organization and contributions

- In Chapter 2, we study the Anderson extrapolation scheme for accelerating the convergence of fixed point methods toward their fixed points. These methods use the solutions to simpler subproblem to perform a weights combination of previous iterates, in order to obtain a better approximation of a fixed point. Anderson acceleration techniques applied to affine mappings enjoy accelerated convergence guarantees. However, moving away from the affine setting can break these convergences results. Modifications of the extrapolation procedure have been proposed in the literature to make the method robust but lack of clear acceleration guarantees. In this chapter, we propose a new version of Anderson acceleration that imposes hard bounds on the extrapolation weights. We provide explicit worst-case guarantees for acceleration in an optimization setting. These bounds rely on the study of a variation of the classical Chebyshev polynomial optimization problem. In this problem, we impose constraints on the coefficients and the polynomial and obtain explicit upper bound on the minimal infinity norm of polynomials over a segment.

  This chapter is associated with the publication:

  - Mathieu Barré, Adrien Taylor, and Alexandre d'Aspremont. "Convergence of constrained Anderson acceleration." arXiv preprint arXiv:2010.15482, 2020 (Submitted).

- Chapter 3 is a introduction to the problem class that are encountered in the next chapters. In addition, this chapter presents the *performance estimation problem* (PEP) methodology that provide a principled way of analyzing fixed-step first-order optimization methods. This part is mainly based on the work of Taylor et al. [2017c] and does not enjoy notable personal contributions. This chapter can be easily skipped by readers familiar with performance estimation.

- Chapter 4 focuses on worst-case analysis of adaptive optimization methods applied to strongly convex functions with Lipschitz gradients. We mainly study algorithms based on the popular Polyak step sizes, which use the optimal value of the objective function. We provide new analyses for various versions of gradient descent with Polyak step sizes, and show that in the worst-case they exhibit similar guarantees as their nonadaptive counterparts. Furthermore, we use these step sizes to develop a momentum based acceleration method. This method demonstrates an acceleration in its linear convergence rate although it does not use the strong convexity parameter.

These convergence guarantees are based on new robustness results on Nesterov's method with constant momentum. On the way, we show how to extend the PEP framework to analyze adaptive algorithms, and we illustrate this approach with numerical worst-cae analyses of several standard adaptive methods.

This chapter is associated with the publication:

- Mathieu Barré, Adrien Taylor, and Alexandre d'Aspremont. "Complexity guarantees for Polyak steps with momentum." Conference on Learning Theory (COLT). PMLR, 2020.

- In Chapter 5, we study optimization methods relying on proximal operations. In particular, we focus on the case of inexact proximal computations, that is, when the subproblem behind the proximal steps are solved approximately. We start by reviewing the different criteria used in the literature to quantify inexactness in proximal operations, and we show that most of them can be cast into a generic formulation. The main contribution of this chapter is to provide a systematic way of analyzing these inexact proximal algorithms, based on the PEP methodology. We illustrate this approach by presenting numerical worst-case bounds for several optimization methods. In addition, we use this methodology to obtain a relatively inexact proximal point algorithm with optimized worst-case guarantees.

  This chapter is associated with the publication:

  - Mathieu Barré, Adrien Taylor, and Francis Bach. "Principled analyses and design of first-order methods with inexact proximal operators." arXiv preprint arXiv:2006.06041, 2020 (Submitted).

- Chapter 6 gathers several inexact proximal algorithms that were obtained and/or analyzed using the methodology of Chapter 5. These results are presented separately as we believe they are of independent interest. In particular, we provide an inexact accelerated forward-backward algorithm possibly supporting strongly convex objectives. This algorithm also allows mixing absolute and relative error terms in proximal computations. In addition, we present a new variant of the *accelerated hybrid proximal extragradient* method [Monteiro and Svaiter, 2013] for possibly strongly convex objective, and analyze a variant of the partially inexact Douglas-Rachford algorithm from Eckstein and Yao [2018].

  This chapter is associated with the publication:

  - Mathieu Barré, Adrien Taylor, and Francis Bach. "A note on approximate accelerated forward-backward methods with absolute and relative errors, and possibly strongly convex objectives." arXiv preprint arXiv:2106.15536, 2021 (Submitted).

**Publications not included in this manuscript**

- Study on a proxy for the sparse recovery threshold in compressed sensing and possible applications in dictionary learning or MRI sampling:

  - Mathieu Barré, and Alexandre d'Aspremont. "An M* Proxy for Sparse Recovery Performance." arXiv preprint arXiv:1810.02748, 2018.

- Detection of methane emitters using Wasserstein barycenter on satellite data:

  - Mathieu Barré, Clément Giron, Matthieu Mazzolini and Alexandre d'Aspremont. "Averaging atmospheric gas concentration data using wasserstein barycenters." arXiv preprint arXiv:2010.02762, 2020.

# Chapter 2

# Constrained Anderson Acceleration

Let $F : \mathbb{R}^d \to \mathbb{R}^d$ be an operator and consider the problem of finding its fixed point, i.e. a solution $x_* \in \mathbb{R}^d$ to

$$x_* = F(x_*). \tag{2.1}$$

When $F$ is a contraction, one can find such a point by running fixed point iterations

$$x_{k+1} = F(x_k)$$

starting from an initial guess $x_0 \in \mathbb{R}^d$. Obtaining faster convergence rates has been a key concern in numerical analysis. Anderson acceleration methods extrapolate a new point hopefully closer to the solution using a linear combination of fixed point iterates $x_k$. This idea was first applied to univariate sequences, fitting a linear model on the iterates and using the fixed point of this model as the extrapolated point [Aitken, 1927, Shanks, 1955, Brezinski, 2006, Brezinski and Redivo-Zaglia, 2019]. Extrapolation techniques were then extended to linearly converging vector valued sequences [Anderson, 1965, Pulay, 1980, Sidi et al., 1986, Smith et al., 1987] with convergence guarantees in the linear case, i.e. when $F$ is an affine operator.

In the nonlinear case (i.e. when $F$ is not an affine operator), convergence results can also be derived using a perturbation argument. However, the weights used to construct extrapolated points are typically obtained by solving ill conditioned quadratic programs, resulting in stability issues. The magnitudes of these weights typically blow up, breaking convergence properties of the acceleration procedure (see examples in e.g. Scieur et al. [2020, Figure 4]). Therefore, convergence guarantees in the nonlinear case mostly rely on additional mechanisms for controlling magnitudes of extrapolation weights.

**Contributions** We study a constrained Anderson acceleration (CAA) algorithm that imposes hard bounds on the extrapolation weights as suggested in Toth and Kelley [2015, Section 2.2] and Scieur et al. [2018], and provide a simple worst-case analysis in a nonlinear setting. We do so by extending the Chebyshev arguments of Scieur et al. [2020] to the constrained case. Overall, our contribution is threefold.

(i) We provide an explicit upper bound for the optimal value of a constrained Chebyshev problem on polynomials. We show this bound is tight on a range of parameters and show numerically that it is close to the optimal value elsewhere.

(ii) We use this bound to construct an explicit, dimension free, worst-case local linear convergence rate for CAA applied to nonlinear operators, and quantify this local acceleration rate.

(iii) We describe an adaptive strategy to adjust the constraints on extrapolation weights, when CAA is applied to a gradient step operator.

**Organization** This chapter starts with an introduction to Chebyshev polynomials and to their role in optimization in Section 2.1. Then, Section 2.2 details the setting considered in this work together with a review of Anderson acceleration results. In Section 2.3, we present the constrained Anderson acceleration algorithm together with its robustness properties. Section 2.4 focuses on the study of a constrained version of the Chebyshev optimization problem. Finally, Section 2.5 is dedicated to Constrained Anderson acceleration applied to the gradient descent method.

**Notations** Depending on the context $\|\cdot\|$ either denotes the classical Euclidean norm (when applied to a vector in $\mathbb{R}^d$), or the operator norm (when applied to a matrix in $\mathbb{R}^{n\times n}$). For $B \subset \mathbb{R}^d$, $\mathrm{diam}(B) = \max_{x,y\in B} \|x-y\|$. We denote by $\mathbf{S}_d^+$ the cone of symmetric positive semidefinite matrices of dimension $d$ and by $Sp(A) \in \mathbb{C}$ the set of eigenvalues of a matrix $A$. For $k \in \mathbb{N}$, $\mathbb{R}_k[X]$ is the vector space of polynomials of degree smaller than $k$, and real coefficients. We denote by $\|\cdot\|_1$ either the sum of the absolute values of the components of a vector (standard $\ell_1$ norm when applied on $\mathbb{R}^n$) or, when applied to a polynomial, the sum of the absolute values of its coefficients. Finally, $I$ denotes the identity operator.

Before diving into the constrained Anderson results, we start by a short introduction to Chebyshev polynomials.

## 2.1 Introduction to Chebyshev polynomials

Let us start with a study some properties of the family of Chebyshev polynomials of first kind (see e.g., Mason and Handscomb [2002] for a large collection of facts on Chebyshev polynomials).

### 2.1.1 Chebyshev polynomials

There exists many equivalent definitions of Chebyshev polynomials, in particular one can use the following recursive one.

**Definition 2.1.1.** *The family of Chebyshev polynomials of first kind* $\{T_k\}_k$ *with* $T_k \in \mathbb{R}_k[X]$ *for* $k \geq 0$ *is defined recursively as*

$$\begin{cases} T_0(X) &= 1 \\ T_1(X) &= X \\ T_{k+2}(X) &= 2XT_{k+1}(X) - T_k(X) \quad \textit{for } k \geq 0. \end{cases}$$

Based on this recursive formulation we can obtain the following property (sometimes directly used to characterize Chebyshev polynomials).

$$T_k(\cos(\theta)) = \cos(k\theta) \quad \text{for all } \sigma \in \mathbb{R} \text{ and } k \geq 0, \tag{2.2}$$

as the cosinus function follows a similar recursive equation

$$\cos((k+2)\theta) = 2\cos(\theta)\cos((k+1)\theta) - \cos(n\theta).$$

Many important properties on Chebyshev polynomials in optimization follow from the fact that they are solutions to polynomial optimization problems described in the next proposition.

**Proposition 2.1.2.** *Let $\alpha \in \mathbb{R}$ with $|\alpha| \geq 1$ and $k \geq 0$,*

$$\frac{T_k(X)}{T_k(\alpha)} = \underset{\substack{p \in \mathbb{R}_k[X] \\ p(\alpha)=1}}{\operatorname{argmin}} \max_{x \in [-1,1]} |p(x)|,$$

*where $T_k$ is the first kind Chebyshev polynomials of degree $k$.*

*Proof.* For $\alpha > 1$, this is the result from Flanders and Shortley [1950, Equation 10].

When $\alpha < 1$, we have that the minimization problem

$$\min_{\substack{p \in \mathbb{R}_k[X] \\ p(\alpha)=1}} \max_{x \in [-1,1]} |p(x)|,$$

is equivalent to

$$\min_{\substack{q \in \mathbb{R}_k[X] \\ q(-\alpha)=1}} \max_{x \in [-1,1]} |q(-x)| \text{ also equivalent to } \min_{\substack{q \in \mathbb{R}_k[X] \\ q(-\alpha)=1}} \max_{x \in [-1,1]} |q(x)|.$$

Therefore using Flanders and Shortley [1950, Equation 10] we get that $T_k(-X)/T_k(-\alpha)$ is the unique solution as $-\alpha > 1$. Finally, we get the desired result by noticing (using e.g. Definition 2.1.1) that $T_{2k}(-X) = T_{2k}(X)$ and $T_{2k+1}(-X) = -T_{2k+1}(X)$ for all $k \geq 0$, implying $T_k(-X)/T_k(-\alpha) = T_k(X)/T_k(\alpha)$. ■

Based on the previous result, let us introduce the rescaled Chebyshev polynomials equal to 1 at 1, that will be used throughout the rest of this chapter.

**Definition 2.1.3.** *Let $a < b < 1 \in \mathbb{R}$ and $k > 0$, we call rescaled Chebyshev polynomial of the first kind, of degree $k$, on $[a, b]$ the polynomial*

$$R_k^{[a,b]}(X) := \frac{T_k(\frac{2(X-a)}{b-a} - 1)}{|T_k(\frac{2(1-a)}{b-a} - 1)|},$$

*where $T_k$ is the Chebyshev polynomial of the first kind, of degree $k$.*

As a consequence of Proposition 2.1.2, we can study the maximum of the absolute value of polynomials over a segment $[a, b]$ with $a < b < 1$ and obtain

**Proposition 2.1.4.** *Let $a < b < 1 \in \mathbb{R}$ and $k \geq 0$,*

$$R_k^{[a,b]}(X) \equiv \frac{T_k\left(\frac{2}{b-a}X - \frac{a+b}{b-a}\right)}{T_k(\frac{2}{b-a} - \frac{a+b}{b-a})} = \underset{\substack{p \in \mathbb{R}_k[X] \\ p(1)=1}}{\operatorname{argmin}} \max_{x \in [a,b]} |p(x)|,$$

*where $T_k$ is the first kind Chebyshev polynomials of degree $k$. In addition*

$$\max_{x \in [a,b]} |R_k^{[a,b]}(x)| = \frac{2\beta^k}{1 + \beta^{2k}},$$

*with* $\beta = \left(\sqrt{1-a} - \sqrt{1-b}\right) / \left(\sqrt{1-a} + \sqrt{1-b}\right)$.

*Proof.* We have

$$\min_{\substack{p\in\mathbb{R}_k[X]\\ p(1)=1}} \max_{x\in[a,b]} |p(x)| = \min_{\substack{p\in\mathbb{R}_k[X]\\ p(1)=1}} \max_{y\in[-1,1]} |p((b-a)\tfrac{y+1}{2}+a)| = \min_{\substack{q\in\mathbb{R}_k[X]\\ q(2\frac{1-a}{b-a}-1)=1}} \max_{y\in[-1,1]} |q(y)|.$$

Thus, if $p_*$ is solution of the left hand side problem, $q_*(y) = p_*((b-a)\frac{y+1}{2}+a)$ is solution of the right hand side one. This last problem is solved using Proposition 2.1.2 with $\alpha = \frac{2(1-a)}{b-a} - 1$ satisfying $|\alpha| > 1$ (using $a < b < 1$). This gives us the solution $q_*(y) = T_k(y)/T_k(\frac{2(1-a)}{b-a} - 1)$, and thus the solution to the original problem $p_*(x) = T_k(2\frac{x-a}{b-a} - 1)/T_k(2\frac{1-a}{b-a} - 1)$.

For the value of the max, we know that $\max_{y\in[-1,1]} |T_k(y)| = 1$, therefore

$$\max_{x\in[0,\rho]} \left| \frac{T_k(\frac{2(X-a)}{b-a}-1)}{T_k(\frac{2(1-a)}{b-a}-1)} \right| = \frac{1}{T_k(\frac{2(1-a)}{b-a}-1)}.$$

Since $\left|\frac{2(1-a)}{b-a} - 1\right| > 1$ one can use the formulas for $T_k(x)$ with $|x| \geq 1$ (see e.g., Mason and Handscomb [2002, Eq 1.49]):

$$T_k(x) = \frac{1}{2}\left(\left(x - \sqrt{x^2-1}\right)^k + \left(x + \sqrt{x^2-1}\right)^k\right) \text{ when } |x| \geq 1.$$

It follows that

$$\begin{aligned}
T_k(\tfrac{2(1-a)}{b-a} - 1) &= \tfrac{1}{2}\left(\left(\tfrac{2-(a+b)}{b-a} - \sqrt{(\tfrac{2-(a+b)}{b-a})^2 - 1}\right)^k + \left(\tfrac{2-(a+b)}{b-a} + \sqrt{(\tfrac{2-(a+b)}{b-a})^2 - 1}\right)^k\right) \\
&= \tfrac{1}{2}\left(\left(\tfrac{2-(a+b)-\sqrt{(2-(a+b))^2-(b-a)^2}}{b-a}\right)^k + \left(\tfrac{2-(a+b)+\sqrt{(2-(a+b))^2-(b-a)^2}}{b-a}\right)^k\right) \\
&= \tfrac{1}{2}\left(\left(\tfrac{2-(a+b)-2\sqrt{(1-a)(1-b)}}{b-a}\right)^k + \left(\tfrac{2-(a+b)+2\sqrt{(1-a)(1-b)}}{b-a}\right)^k\right) \\
&= \tfrac{1}{2}\left(\left(\tfrac{\left(\sqrt{1-a}-\sqrt{1-b}\right)^2}{b-a}\right)^k + \left(\tfrac{\left(\sqrt{1-a}+\sqrt{1-b}\right)^2}{b-a}\right)^k\right) \\
&= \tfrac{\left(\sqrt{1-a}+\sqrt{1-b}\right)^{2k}}{2(b-a)^k}\left(1 + \left(\tfrac{\sqrt{1-a}-\sqrt{1-b}}{\sqrt{1-a}+\sqrt{1-b}}\right)^{2k}\right) \\
&= \tfrac{\left(\sqrt{1-a}+\sqrt{1-b}\right)^{k}}{2\left(\sqrt{1-a}-\sqrt{1-b}\right)^{k}}\left(1 + \left(\tfrac{\sqrt{1-a}-\sqrt{1-b}}{\sqrt{1-a}+\sqrt{1-b}}\right)^{2k}\right),
\end{aligned}$$

providing the desired result. ∎

Later on this chapter we focus on a variation of the optimization problem in Proposition 2.1.4, with additional constraints on the coefficients of the polynomials, which makes the nice Chebyshev argument impossible to apply.

Based on Proposition 2.1.4, we can finally state an important result that is at the center of many worst-case analyses of convex quadratic minimization algorithms.

**Theorem 2.1.5.** *Let $d \in \mathbb{N}^*$, $G \in \mathbf{S}_d^+$ such that $\mathrm{Sp}(G) \subset [0, \rho]$ with $0 < \rho < 1$, it holds that*

$$\forall z \in \mathbb{R}^d, \min_{\substack{p\in\mathbb{R}_k[X]\\ p(1)=1}} \|p(G)z\| \leq \frac{2\beta^k}{1+\beta^{2k}}\|z\|,$$

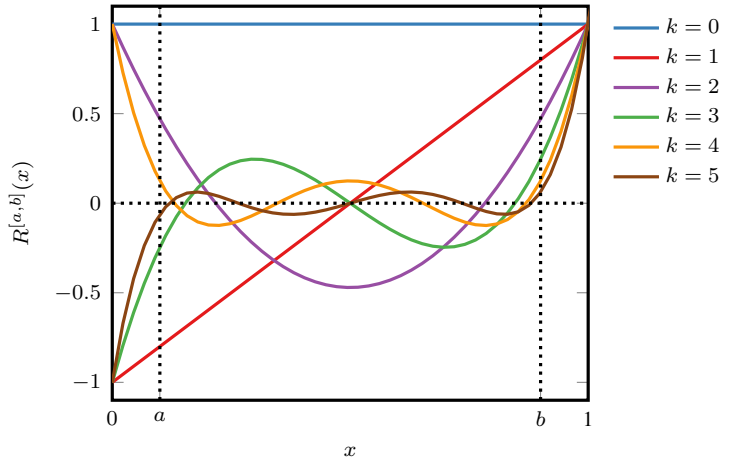


Figure 2.1: Representation of $R_k^{[a,b]}(X)$ on the segment $[a, b]$ with $a = 0.1$ and $b = 0.9$ for various values of $k$.

*with* $\beta = (1 - \sqrt{1-\rho}) \,/\, (1 + \sqrt{1-\rho})$.

*Proof.* Since $G$ is positive definite, it can be expressed as $ODO^T$, with $O$ an orthogonal matrix and $D$ a diagonal matrix with $\mathrm{Sp}(D) \subset [0, \rho]$. We have

$$\begin{aligned} \|p(G)\| &= \|OP(D)O^T\| \\ &\leq \|p(D)\| = \max_{x \in \mathrm{Sp}(D)} |p(x)| \\ &\leq \max_{x \in [0,\rho]} |p(x)|, \end{aligned}$$

and applying Proposition 2.1.4 with $a = 0$ and $b = \rho$ leads to the desired conclusion. ∎

The following reformulation can be more practical depending on the application.

**Theorem 2.1.6.** *Let $d \in \mathbb{N}^*$, $A \in \mathbf{S}_d^+$ such that $\mathrm{Sp}(A) \subset [\mu, L]$ with $0 < \mu < L$, it holds that*

$$\forall z \in \mathbb{R}^d, \quad \min_{\substack{p \in \mathbb{R}_k[X] \\ p(0)=1}} \|p(A)z\| \leq \frac{2\beta^k}{1+\beta^{2k}} \|z\|,$$

*with* $\beta = \left(\sqrt{L} - \sqrt{\mu}\right) \Big/ \left(\sqrt{L} + \sqrt{\mu}\right)$.

*Proof.* It simply comes from the reformulation

$$\min_{\substack{p \in \mathbb{R}_k[X] \\ p(0)=1}} \|p(A)z\| = \min_{\substack{q \in \mathbb{R}_k[X] \\ q(1)=1}} \|q(I - \tfrac{1}{L}A)z\|,$$

which allows applying Theorem 2.1.5 to $G = I - \frac{1}{L}A$ with $\mathrm{Sp}(G) \subset [0, 1 - \frac{\mu}{L}]$. ∎

These theorems are used to provide convergence guarantees for many optimization algorithms such that the conjugate gradient method (see textbooks Shewchuk [1994], Nocedal and Wright [2006]) or GMRES [Saad and Schultz, 1986].

In the following, we illustrate these Chebyshev results on the study the celebrated Chebyshev iterations (or Chebyshev acceleration).

### 2.1.2 Application: Chebyshev iterations

In this section, we detail an application of Theorem 2.1.5 in optimization. Given an affine contractive operator $F : \mathbb{R}^d \to \mathbb{R}^d$ with unique fixed point $x_*$. The standard fixed point iteration writes as

$$x_{k+1} = F(x_k),$$

and converges toward the fixed point $x_*$ for all initialization $x_0 \in \mathbb{R}^d$. The convergence of this fixed point iterations method can be slow, with a linear convergence rate in $\rho < 1$ when $F$ is $\rho$-contractive (i.e. $\rho$-Lipschitz). That is,

$$\|x_k - x_*\| \leq \rho^k \|x_0 - x_*\|.$$

When $\rho$ is known, and $F$ is affine such that $F(x) = x_* + G(x - x_*)$ with $G \in \mathbf{S}_d^+$ and $\mathrm{Sp}(G) \subset [0, \rho]$, a standard technique to accelerate the convergence consists in using the so called Chebyshev iterations (see e.g., Golub and Varga [1961], Varga [1962], Nemirovsky [1992]). This method aims at exploiting the results of Proposition 2.1.4 and Theorem 2.1.5 to obtain that

$$\|x_k - x_*\| \leq \rho_*(k) \|x_0 - x_*\|,$$

where $\rho_*(k) = 2\beta^k/(1+\beta^{2k}) < \rho^k$ and $\beta = (1-\sqrt{1-\rho})/(1+\sqrt{1-\rho})$. The Chebyshev acceleration method corresponds to chose iterates $x_k$ such that the residual $x_k - x_*$ satisfies

$$x_k - x_* = R^{[0,\rho]}(G)(x_0 - x_*) \quad \text{for } k \geq 1, \tag{2.3}$$

where $R^{[0,\rho]}(X)$ is defined in Definition 2.1.3.

Using the recursion formula of Definition 2.1.1 allows obtaining a more practical form for the updates. Indeed, it holds that

$$T_{k+2}\left(\tfrac{2}{\rho}X - 1\right) = 2\left(\tfrac{2}{\rho}X - 1\right) T_{k+1}\left(\tfrac{2}{\rho}X - 1\right) - T_k\left(\tfrac{2}{\rho}X - 1\right) \ \forall k \geq 0,$$

which leads to the iterates expression

$$\begin{aligned}
x_{k+2} &= x_* + \frac{2\left(\frac{2}{\rho}G-1\right)T_{k+1}\left(\frac{2}{\rho}G-1\right) - T_k\left(\frac{2}{\rho}G-1\right)}{T_{k+2}\left(\frac{2}{\rho}-1\right)}(x_0 - x_*) \\
&= x_* + \frac{2\left(\frac{2}{\rho}G-1\right)T_{k+1}\left(\frac{2}{\rho}-1\right)}{T_{k+2}\left(\frac{2}{\rho}-1\right)}(x_{k+1} - x_*) - \frac{T_k\left(\frac{2}{\rho}-1\right)}{T_{k+2}\left(\frac{2}{\rho}-1\right)}(x_k - x_*) \\
&= \frac{4T_{k+1}\left(\frac{2}{\rho}-1\right)}{\rho T_{k+2}\left(\frac{2}{\rho}-1\right)}(x_* + G(x_{k+1} - x_*)) - 2\frac{T_{k+1}\left(\frac{2}{\rho}-1\right)}{T_{k+2}\left(\frac{2}{\rho}-1\right)}x_{k+1} - \frac{T_k\left(\frac{2}{\rho}-1\right)}{T_{k+2}\left(\frac{2}{\rho}-1\right)}x_k,
\end{aligned}$$

where in the last line we used $T_{k+2}\left(\frac{2}{\rho} - 1\right) = 2\left(\frac{2}{\rho} - 1\right) T_{k+1}\left(\frac{2}{\rho} - 1\right) - T_k\left(\frac{2}{\rho} - 1\right)$. Finally, using $R_1^{[0,\rho]}(X) = \frac{2}{2-\rho}X - \frac{\rho}{2-\rho}$ we can reformulate (2.3) as

$$\left\{\begin{array}{rcl}
x_0 & \in & \mathbb{R}^d \\
x_1 & = & \frac{2}{2-\rho}(x_* + G(x_0 - x_*)) - \frac{\rho}{2-\rho}x_0 \\
x_{k+2} & = & \frac{4\omega_{k+1}}{\rho}(x_* + G(x_{k+1} - x_*)) - 2\omega_{k+1}x_{k+1} - \left(2(\frac{2}{\rho} - 1)\omega_{k+1} - 1\right)x_k, \\
 & & \text{for } k \geq 0, \text{ where } \omega_{k+1} = -\dfrac{T_{k+1}\left(\frac{2}{\rho}-1\right)}{T_{k+2}\left(\frac{2}{\rho}-1\right)}.
\end{array}\right. \tag{2.4}$$

We have seen in the proof of Theorem 2.1.6 that

$$T_k\left(\frac{2}{\rho}-1\right)=\frac{1+\beta^{2k}}{2\beta^k},\ \text{with}\ \beta=\frac{1-\sqrt{1-\rho}}{1+\sqrt{1-\rho}},$$

which leads to the simplified expression

$$\omega_{k+1}=\beta\frac{1+\beta^{2(k+1)}}{1+\beta^{2(k+2)}},\ \text{with}\ \beta=\frac{1-\sqrt{1-\rho}}{1+\sqrt{1-\rho}}.$$

Finally, we can express (2.3) as:

**Chebyshev acceleration**

**Input:**

- Operator $F:\mathbb{R}^d\to\mathbb{R}$.
- Initial guess: $x_0\in\mathbb{R}^d$.
- Contraction factor: $0<\rho<1$ .

**Initialization:**

$$x_1=\frac{2}{2-\rho}F(x_0)-\frac{\rho}{2-\rho}x_0,$$

**Run:**

For $k=1,\dots$ :

$$\begin{aligned}\omega_k&=\beta\frac{1+\beta^{2k}}{1+\beta^{2(k+1)}}\\ x_{k+1}&=\frac{4\omega_k}{\rho}F(x_{k+1})-2\omega_k x_{k+1}-\left(2(\frac{2}{\rho}-1)\omega_k-1\right)x_k\end{aligned}\qquad\text{(Cheb-acc)}$$

End For

**Output:** $x_{k+1}$

**Remark 2.1.7.** *Chebyshev acceleration can be reformulated in an optimization setting, where we seek to minimize a convex quadratic function $f$ defined as*

$$f(x)=\tfrac{1}{2}(x-x_*)^TA(x-x_*)+f_*,$$

*with $A\in\mathbf{S}_d^+$ such that $\mathrm{Sp}(A)\subset[\mu,L]$, $(0<\mu<L)$, $x_*\in\mathbb{R}^d$ and $f_*\in\mathbb{R}$. In that case, we can apply Chebyshev iterations to the contractive operator $F=I-\frac{1}{L}\nabla f$ with contraction factor $\rho=1-\frac{\mu}{L}$.*

**Remark 2.1.8.** *Since $0<\beta<1$ one notices that $\omega_k$ tends to $\beta$ with the iterations. A variation of Chebyshev acceleration consists in fixing $\omega_k$ equal to $\beta$. This method is closely related to Polyak's heavy-ball method in optimization [Polyak, 1964].*

As announced at the beginning of this section, we can get the following worst-case bound on the distance to the optimum.

**Proposition 2.1.9.** *Let $d \in \mathbb{N}^*$, $F : \mathbb{R}^d \to \mathbb{R}^d$ be a contractive operator such that $F(x) = x_* + G(x - x_*)$ with $\mathrm{Sp}(G) \subset [0, \rho]$, $0 < \rho < 1$ and $x_* \in \mathbb{R}^d$. The iterates of the Chebyshev acceleration method (Cheb-acc) applied on $F$ and initiated at $x_0 \in \mathbb{R}^d$ with contraction parameter $\rho$ satisfy*

$$\|x_k - x_*\| \leq \frac{2\beta^k}{1 + \beta^{2k}} \|x_0 - x_*\|, \text{ for } k \geq 0,$$

*with $\beta = (1 - \sqrt{1-\rho})/(1 + \sqrt{1-\rho})$ and $F(x_*) = x_*$.*

*Proof.* Similar to the proof of Theorem 2.1.5,

$$\|x_k - x_*\| = \|R_k^{[0,\rho]}(G)(x_0 - x_*)\| \leq \|R_k^{[0,\rho]}\| \|x_0 - x_*\|,$$

and we conclude using Proposition 2.1.4. ∎

We illustrate on Figure 2.2, how the Chebyshev acceleration method can accelerate convergence toward a fixed point of an affine mapping compared with a simple fixed point procedure.

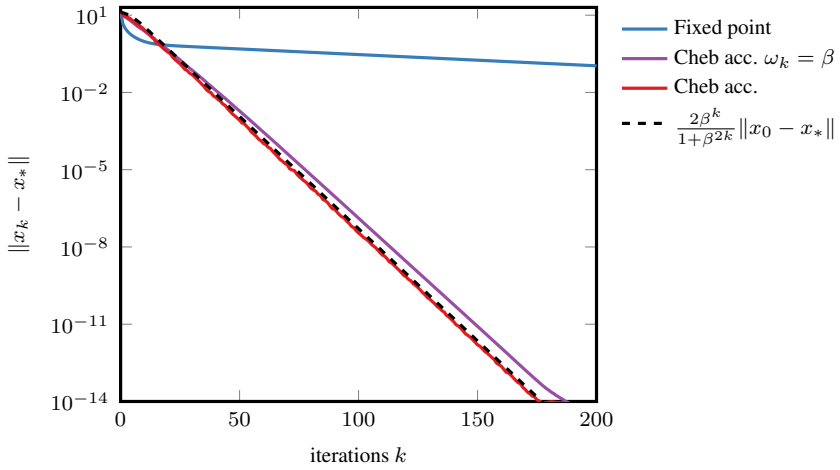


Figure 2.2: Numerical comparison of fixed point iterations (blue), Chebyshev acceleration with constant $\omega_k = \beta$ (purple) and Chebyshev acceleration (red), on an affine contractive mapping with $\rho = 0.99$. The black dashed curve corresponds to the bound of Proposition 2.1.9 with $\beta = (1 - \sqrt{1-\rho}) \,/\, (1 + \sqrt{1-\rho})$.

**Remark 2.1.10.** *In order for Proposition 2.1.9 to hold, the contraction factor of the affine mapping $F$ has to be known in advance (or an upper-bound of it). This constitute an important limitation in the application of Chebyshev acceleration as this parameter is difficult to estimate in many situations (e.g., estimation of strong convexity parameter in optimization).*

In this section, we reviewed some elements of numerical analysis based on Chebyshev polynomials. In the rest of the chapter we focus on Anderson acceleration and on a variation of the polynomial problem appearing in Proposition 2.1.4 and Theorem 2.1.5.

## 2.2 Preliminaries

From this point on, we study the linear convergence of a constrained Anderson acceleration scheme on an operator $F : \mathbb{R}^d \to \mathbb{R}^d$. In recent applications of Anderson extrapolation in optimization, $F$ is typically a gradient step with constant step size (e.g.,Scieur et al. [2020], Mai and Johansson [2020]). We use two types of assumptions on $F$ throughout.

**Assumption 2.2.1.** *$F$ is $\rho$-Lipschitz with $\rho < 1$, and can be decomposed as*

$$F = G + \xi$$

*for a linear $G \in \mathbf{S}_d^+$ with $G \preccurlyeq \rho I$ and a nonlinear $\xi : \mathbb{R}^d \to \mathbb{R}^d$ $\alpha$-Lipschitz with $\alpha \geq 0$.*

**Assumption 2.2.2.** *$F$ is $\rho$-Lipschitz with $\rho < 1$ and is continuously differentiable with positive semidefinite and $\eta$-Lipschitz Jacobian $F'$ where $\eta \geq 0$.*

The $\rho$-Lipschitzness assumption implies that $F$ has a unique fixed point $x_*$ and the iterates of the fixed point iterations $x_{k+1} = F(x_k)$ satisfy $\|x_{k+1} - x_k\| \leq \rho\|x_k - x_{k-1}\|$ and $x_k \to x_*$. The second assumption implies that for $x_0 \in \mathbb{R}^d$ and a compact set $B \subset \mathbb{R}^d$ containing $x_0$, one can decompose $F$ as $F = G + \xi$ with $G = F'(x_0)$ and $\xi = F - F'(x_0)$. Moreover $\xi$ is locally Lipschitz over $B$ with Lispchitz constant roughly equal to $\eta \operatorname{diam}(B)$, decreasing with the diameter of $B$. Note that Assumption 2.2.2 does not enforce Assumption 2.2.1 to hold as it implies local Lipschitzness.

**Remark 2.2.3.** *We illustrate these assumptions in the optimization setting.*

- *When $f : \mathbb{R}^d \to \mathbb{R}$ is a quadratic function with $\mu I \preceq \nabla^2 f \preceq LI$ and $0 < \mu \leq L$, the gradient step operator $F = I - \frac{1}{L}\nabla f$ is affine and satisfies Assumption 2.2.1 with $\rho = \left(1 - \frac{\mu}{L}\right)$ and $\xi = 0$.*
- *When $f : \mathbb{R}^d \to \mathbb{R}$ is a $\mathcal{C}^2$, $\mu$-strongly convex function with $L$-Lipschitz gradient for $0 < \mu \leq L$, and $\eta$-Lipschitz Hessian, the gradient step operator $F = I - \frac{1}{L}\nabla f$ is nonlinear and satisfies Assumption 2.2.2 with $\rho = \left(1 - \frac{\mu}{L}\right)$ (see e.g., Ryu and Boyd [2016]).*

We focus on a fixed depth version of Anderson acceleration. For a predetermined constant $k \in \mathbb{N}$, this simple method consists in performing $k+1$ fixed point iterations with $F$ and use these $k+1$ iterates to get an extrapolated point. We can then restart the method at the extrapolated point. The extrapolated solution is obtained by solving a quadratic program with a bound on the $\ell_1$ norm of extrapolation weights. This choice of norm is motivated by a tightness result derived in Theorem 2.4.8, but any norm would lead to similar developments. The procedure is described in Algorithm 2.2.

In practice, $k$ is set to a small constant (e.g. 5 or 10) and Algorithm 2.2 is restarted by plugging the extrapolated output as input (see e.g., Scieur et al. [2018]) for a new run of the method. The linearly constrained quadratic subproblem in (2.5) for computing the extrapolation weights is low dimensional and can be easily solved by e.g., interior-point methods.

We look at convergence bounds of the form $\|F(x_e) - x_e\| \leq \tilde{\rho}\|F(x_0) - x_0\|$, where $x_e$ is the output of Algorithm 2.2 started at $x_0$ and the quantity $\|F(x) - x\|$ controls how far $x$ is from being a fixed point of $F$. This choice allows to chain together the convergence guarantees for consecutive run of Algorithm 2.2. When $F$ satisfies Assumption 2.2.1 we always have that $\|F(x_k) - x_k\| \leq \rho^k\|F(x_0) - x_0\|$ hence we consider that extrapolation provides convergence acceleration as soon as $\tilde{\rho} < \rho^k$

**Constrained Anderson acceleration** (Algorithm 2.2)

**Input:**

- Contractive operator: $F : \mathbb{R}^d \to \mathbb{R}^d$.
- Initial guess: $x_0 \in \mathbb{R}^d$.
- Bound on extrapolation weights: $C \geq 1$.
- Number of iterates used in extrapolation: $k \in \mathbb{N}^*$.

**Run:**

For $i = 0, \ldots, k$:

$$x_{i+1} = F(x_i)$$

End For

Form $R = \begin{bmatrix} x_0 - x_1 & \cdots & x_k - x_{k+1} \end{bmatrix}$ and compute

$$\tilde{c} = \underset{\mathbf{1}^T c = 1,\ \|c\|_1 \leq C}{\operatorname{argmin}} \|Rc\| \tag{2.5}$$

**Output:** Extrapolated point $x_e = \sum_{i=0}^{k} \tilde{c}_i x_i$

**Related work** Several recent results have been focused on improving convergence guarantees for acceleration methods. In Scieur et al. [2020], the authors apply a regularized formulation of Anderson extrapolation in an optimization setting. Regularization yields accelerated linear convergence rates in some asymptotic regimes, without any additional hypothesis on the independence of the residuals. Brezinski et al. [2020] also proposes a stabilized version guaranteeing local linear acceleration without any linear independence hypothesis but with an assumption on the conditioning of the Jacobian.

Acceleration mechanisms, and Anderson acceleration in particular, have a strong link with quasi-Newton methods Fang and Saad [2009], Rohwedder and Schneider [2011]. A variant of Anderson acceleration called the DIIS procedure has been studied in Rohwedder and Schneider [2011] and yields accelerated local linear convergence under a linear independence hypothesis on differences of consecutive residuals and an hypothesis on the conditioning of the Jacobian of $I - F$ at a fixed point $x_*$. The idea of imposing a sufficient linear independence condition on the difference of the residuals is also present in Pollock and Rebholz [2019]. It has also been shown in Toth and Kelley [2015] that when the extrapolation weights are bounded, AA is locally linearly convergent. However, none of these conditions guarantee *a priori* improved linear convergence rates, as they are impossible to check without actually running the method.

A globally convergent modification of the DIIS procedure is proposed in Chen and Kelley [2019] consisting in using only positive weights in the extrapolation. However, using only positive weights amounts to forming convex combination of previous iterates which severely limits acceleration. An adaptive regularization scheme in Ouyang et al. [2020] provides acceleration guarantees under boundedness hypothesis on the extrapolation weights, extending the work of Toth and Kelley [2015]. A globally converging Anderson acceleration type algorithm is also presented in Zhang et al. [2020];

however, due to the very general assumptions made in the paper, no convergence rate is provided. In Chupin et al. [2020], an adaptive restart strategy yields local superlinear convergence without any assumption on conditioning, but the region around the optimum where superlinear convergence occurs is dependent on the ambient dimension $d$ and its size goes to 0 when $d$ tends to infinity.

Finally, these extrapolation methods were widely extended in the optimization community: to the stochastic setting [Scieur et al., 2017], to composite optimization problems [Massias et al., 2018, Mai and Johansson, 2020], to splitting methods [Poon and Liang, 2019, Fu et al., 2020], to coordinate descent [Bertrand and Massias, 2021] and to accelerate momentum based methods [Bollapragada et al., 2019].

The setting of this chapter is essentially that of Scieur et al. [2020], which is more restrictive than those of Pollock and Rebholz [2019], Ouyang et al. [2020], Brezinski et al. [2020], Chupin et al. [2020] (in particular because of the symmetry assumption on $G$). This setup allows proving explicit, dimension independent, worst-case local linear convergence rates, a priori, without additional assumption on the iterates themselves, or on the optimum.

## 2.3 Constrained Anderson acceleration

We first recall some standard results on Anderson acceleration on linear operators when $\alpha = 0$ in Assumption 2.2.1 (or $\eta = 0$ in Assumption 2.2.2). We then introduce constraints on the extrapolation coefficients for stabilizing the extrapolation procedure, and deal with nonlinearity through the introduction of perturbation parameters $\alpha > 0$ in Assumption 2.2.1 (or $\eta > 0$ in Assumption 2.2.2).

### 2.3.1 Anderson acceleration on linear problems

Let us consider the case $\alpha = 0$ (i.e., $F$ is affine), where Algorithm 2.2 can be used with $C = \infty$. We recall the well-known convergence result on Anderson acceleration in the linear case.

**Proposition 2.3.1.** *Let $F$ be satisfying Assumption 2.2.1 with $\alpha = 0$, $x_e \in \mathbb{R}^d$ be the output of Algorithm 2.2 initiated at some $x_0 \in \mathbb{R}^d$ such that $F(x_0) \neq x_0$, and let $C = \infty$ and $k > 0$. We have that*

$$\frac{\|F(x_e) - x_e\|}{\|F(x_0) - x_0\|} \leq \min_{\substack{p \in \mathbb{R}_k[X] \\ p(1)=1}} \max_{x \in [0,\rho]} |p(x)| = \rho_* := \frac{2\beta^k}{1+\beta^{2k}},$$

*with $\beta = \frac{1-\sqrt{1-\rho}}{1+\sqrt{1-\rho}}$. In addition $\rho_* < \rho^k$.*

*Proof.* Reformulation of Scieur et al. [2020, Proposition 2.1]. ∎

In the following, $\alpha$ or $\eta$ may be nonzero and the previous proposition does not apply.

### 2.3.2 Constrained Anderson acceleration on nonlinear problems

When applying the extrapolation step (2.5) to a nonlinear operator $F$, the conditioning of the matrix $R^T R \in \mathbf{S}^+_{k+1}$ becomes an important issue. This matrix might be singular in some particular situations (see below), but more importantly becomes very close to singular in typical situations. For instance, when $F$ is a gradient step operator of a smooth convex function, consecutive gradients tend to get aligned (in particular, this can be easily formalized when $F$ is the gradient step operator of a convex quadratic function), leading to a very ill-conditioned $R^T R$ in that case. Furthermore, if $F$ is the gradient of a convex quadratic function with Hessian $H \succeq 0$, and $x_0$ is an eigenvector of $H$, the matrix $R^T R$

will be singular. This means the solution vector $\tilde{c}$ can have coefficients with very large magnitude. When $\alpha > 0$, those coefficients are multiplied with the nonlinear part of $F$ and can make the iterates of the algorithm diverge (see Scieur et al. [2020] for an example of such divergence). A solution to fix this issue is to check the conditioning of the matrix (or some related quantity) and adjust iterations depending on it (e.g. restart [Chupin et al., 2020] or discard iterates [Brezinski et al., 2020]). A more direct method consists in controlling the magnitude of these coefficients, by e.g. regularizing (2.5), as in Scieur et al. [2020] (with $C = \infty$), or by imposing hard constraints on $\tilde{c}$, as we do here. Whereas regularization renders computations easier in practice, imposing constraints makes the analysis simpler.

**Proposition 2.3.2.** *Let $F$ be an operator satisfying Assumption 2.2.1, $\alpha \geq 0$ and $x_e \in \mathbb{R}^d$ be the output of Algorithm 2.2 initiated at $x_0 \in \mathbb{R}^d$ with $C \geq 1$ and $k \geq 1$. We have*

$$\|F(x_e) - x_e\| \leq \left( \max_{x \in [0,\rho]} |p_*^C(x)| + 3C\alpha k \right) \|F(x_0) - x_0\|, \tag{2.6}$$

*where*

$$p_*^C \in \operatorname*{argmin}_{\substack{p \in \mathbb{R}_k[X] \\ p(1)=1 \\ \|p\|_1 \leq C}} \max_{x \in [0,\rho]} |p(x)|,$$

*and $\|p\|_1$ is the $\ell_1$ norm of the vector of coefficients of $p$. In addition, under Assumption 2.2.2, the bound in (2.6) holds with $\alpha = kC\eta\|F(x_0) - x_0\|$.*

*Proof.* The proof mostly relies on reformulations and triangle inequalities. We consider fixed point iterations of $F$, of the form

$$x_{i+1} = G(x_i) + \xi(x_i).$$

Equivalently, such iterations can be described as

$$x_{i+1} - x_* = G(x_i - x_*) + \xi(x_i) - \xi(x_*),$$

with $x_* = F(x_*)$. Expanding previous expression, one can rewrite the iterative process as

$$x_{i+1} - x_* = G^{i+1}(x_0 - x_*) + \sum_{j=0}^{k} G^{i-j}(\xi(x_j) - \xi(x_*)),$$

or in terms of the fixed point residual $F(x_i) - x_i = x_{i+1} - x_i$,

$$x_{i+1} - x_i = (G - I)G^i(x_0 - x_*) + (G - I)\sum_{j=0}^{i-1} G^{i-j-1}\left(\xi(x_j) - \xi(x_*)\right) + \xi(x_i) - \xi(x_*).$$

Let us use those expressions, along with a triangle inequality, to work out the fixed point residual after extrapolation

$$\begin{aligned}
&\|(F - I)(x_e)\| \\
=&\|(G - I)(x_e - x_*) + \xi(x_e) - \xi(x_*)\| \\
=&\left\| \sum_{i=0}^{k} \tilde{c}_i (G - I)(x_i - x_*) + \xi(x_e) - \xi(x_*) \right\|
\end{aligned}$$

$$= \left\| \sum_{i=0}^{k} \tilde{c}_i G^i (G-I)(x_0 - x_*) + (G-I) \sum_{i=0}^{k} \tilde{c}_i \sum_{j=0}^{i-1} G^{i-1-j}(\xi(x_j) - \xi(x_*)) + \xi(x_e) - \xi(x_*) \right\|$$
$$= \left\| \sum_{i=0}^{k} \tilde{c}_i (x_{i+1} - x_i) - \sum_{i=0}^{k} \tilde{c}_i \xi(x_i) + \xi(x_e) \right\|,$$

where we used $\sum_{i=0}^{k} \tilde{c}_i = 1$ in the last step. We finally arrive to

$$\|(F-I)(x_e)\| \leq \left\| \sum_{i=0}^{k} \tilde{c}_i (x_{i+1} - x_i) \right\| + \left\| \xi(x_e) - \sum_{i=0}^{k} \tilde{c}_i \xi(x_i) \right\|, \tag{2.7}$$

where the first term on the right hand side is exactly the quantity that is minimized in Algorithm 2.2. We then bound the two terms separately. Let $c_*$ denotes the coefficients of the polynomial $p_*^C = \underset{\substack{p \in \mathbb{R}_k[X] \\ p(1)=1 \\ \|p\|_1 \leq C}}{\operatorname{argmin}} \max_{x \in [0,\rho]} |p(x)|$, we proceed as follows:

$$\left\| \sum_{i=0}^{k} \tilde{c}_i (x_{i+1} - x_i) \right\| \leq \left\| \sum_{i=0}^{k} c_i^* (x_{i+1} - x_i) \right\| \text{ since } c_* \text{ feasible for problem (2.5)},$$

and then

$$\begin{aligned}
&\left\| \sum_{i=0}^{k} c_i^* (x_{i+1} - x_i) \right\| \\
&= \left\| \sum_{i=0}^{k} c_i^* G^i (G-I)(x_0 - x_*) + (G-I) \sum_{i=0}^{k} c_i^* \sum_{j=0}^{i-1} G^{i-1-j}(\xi(x_j) - \xi(x_*)) \right. \\
&\quad \left. + \sum_{i=0}^{k} c_i^* (\xi(x_i) - \xi(x_*)) \right\| \\
&= \left\| \sum_{i=0}^{k} c_i^* G^i \left[ (G-I)(x_0 - x_*) + \xi(x_0) - \xi(x_*) \right] + (G-I) \sum_{i=0}^{k} c_i^* \sum_{j=0}^{i-1} G^{i-1-j}(\xi(x_j) - \xi(x_*)) \right. \\
&\quad \left. + \sum_{i=0}^{k} c_i^* \left[ \xi(x_i) - \xi(x_*) - G^i(\xi(x_0) - \xi(x_*)) \right] \right\| \\
&\leq \left\| \sum_{i=0}^{k} c_i^* G^i \left[ (F-I)(x_0) \right] \right\| \\
&\quad + \left\| (G-I) \sum_{i=1}^{k} c_i^* \sum_{j=0}^{i-1} G^{i-1-j}(\xi(x_j) - \xi(x_*)) + \sum_{i=1}^{k} c_i^* \left[ \xi(x_i) - \xi(x_*) - G^i(\xi(x_0) - \xi(x_*)) \right] \right\| \\
&\leq \left\| p_*^C(G) \right\| \|(F-I)(x_0)\| + \left\| \sum_{i=1}^{k} c_i^* \left[ \sum_{j=1}^{i} G^{i-j}(\xi(x_j) - \xi(x_*)) - \sum_{j=0}^{i-1} G^{i-1-j}(\xi(x_j) - \xi(x_*)) \right] \right\| \\
&\leq \left\| p_*^C(G) \right\| \|(F-I)(x_0)\| + \left\| \sum_{i=1}^{k} c_i^* \sum_{j=0}^{i-1} G^{i-j-1} \left[ \xi(x_{j+1}) - \xi(x_j) \right] \right\|
\end{aligned}$$

$$
\begin{aligned}
&\leq \left\|p_*^C(G)\right\| \|(F-I)(x_0)\| + \alpha \sum_{i=1}^{k} |c_i^*| \sum_{j=0}^{i-1} \rho^{i-j-1}\rho^j \|(F-I)(x_0)\| \\
&\leq \left(\left\|p_*^C(G)\right\| + \alpha \sum_{i=1}^{k} |c_i^*| \rho^{i-1} i\right) \|(F-I)(x_0)\| \\
&\leq \left(\left\|p_*^C(G)\right\| + \alpha k \|c^*\|_1\right) \|(F-I)(x_0)\| \\
&\leq \left(\left\|p_*^C(G)\right\| + \alpha k C\right) \|(F-I)(x_0)\| .
\end{aligned}
$$

One can bound $\|p_*^C(G)\|$ with standard arguments: Since $0 \preccurlyeq G \preccurlyeq \rho I$, there exist an orthogonal matrix $O$ and a diagonal matrix $D$ such that $G = O^t D O$. Therefore, we get $\|p_*(G)\| = \|O^t p_*^C(D) O\| \leq \|p_*^C(D)\|$. One can then notice that $\|p_*^C(D)\| = \max_{\lambda \in Sp(G)} |p_*^C(\lambda)| \leq \max_{x \in [0,\rho]} |p_*^C(x)|$, where $Sp(G)$ is the set of eigenvalues of $G$. Let us bound the second term of the right hand side in (2.7)

$$
\begin{aligned}
\|\xi(x_e) - \sum_{i=0}^{k} \tilde{c}_i \xi(x_i)\| &\leq \|\xi(x_e) - \xi(x_k)\| + \|\xi(x_k) - \sum_{i=0}^{k} \tilde{c}_i \xi(x_i)\| \\
&\leq \alpha \left( \|x_e - x_k\| + \sum_{i=0}^{k} |\tilde{c}_i| \|x_k - x_i\| \right) \\
&\leq 2\alpha \sum_{i=0}^{k-1} |\tilde{c}_i| \|x_k - x_i\| \\
&\leq 2\alpha \sum_{i=0}^{k-1} |\tilde{c}_i| \rho^i \|x_{k-i} - x_0\| \\
&\leq 2\alpha \sum_{i=0}^{k-1} |\tilde{c}_i| \rho^i \sum_{j=0}^{k-1-i} \|x_{j+1} - x_j\| \\
&\leq 2\alpha \sum_{i=0}^{k-1} |\tilde{c}_i| \rho^i (k-i) \|(F-I)(x_0)\| \\
&\leq 2\alpha k \|\tilde{c}\|_1 \|(F-I)(x_0)\| \\
&\leq 2\alpha k C \|(F-I)(x_0)\|.
\end{aligned}
$$

Combining the two previous bounds allows reaching (2.6).

Let Assumption 2.2.2 hold, we can then pick $G = F'(x_0)$ and $\xi = F - F'(x_0)$ (note that $F'(x_0)$ is symmetric positive semidefinite by assumption and that $\|F'(x_0)\| \leq \rho$ when $F$ is $\rho$-Lipschitz). In computations of the previous bounds, Lischitzness was only used on the convex set $B_C = \{\sum_{i=0}^{k} c_i x_i : \|c\|_1 \leq C, \sum_{i=0}^{k} c_i = 1\}$. Let us bound $\|D\xi(x)\|$ for $x = \sum_{i=0}^{k} c_i x_i$ in $B_C$.

$$
\begin{aligned}
\|D\xi(x)\| &= \|F'(x) - F'(x_0)\| \\
&\leq \eta \|x - x_0\| = \eta \left\| \sum_{i=0}^{k} c_i (x_i - x_0) \right\| \\
&\leq \eta \sum_{i=1}^{k} |c_i| \left\| \sum_{j=0}^{i-1} x_{j+1} - x_j \right\|
\end{aligned}
$$

$$\leq \eta \sum_{i=1}^{k} |c_i| \sum_{j=0}^{i-1} \rho^j \|x_1 - x_0\|$$
$$\leq \eta C k \|F(x_0) - x_0\| .$$

Using the mean value theorem, we conclude that $\xi(x)$ is $\eta C k \|F(x_0) - x_0\|$-Lipschitz on $B_C$.

■

The following corollary simply states that one can allow a small relative error in the computation of (2.5) in Algorithm 2.2 while keeping linear convergence.

**Corollary 2.3.3.** *Let $F$ be an operator satisfying Assumption 2.2.1, $\alpha \geq 0$ and $x_e \in \mathbb{R}^d$ be the output of Algorithm 2.2 initiated at $x_0 \in \mathbb{R}^d$ with $C \geq 1$ and $k \geq 1$. If (2.5) is solved with relative precision $\varepsilon\|F(x_0) - x_0\|$ on its optimality gap for some $\varepsilon > 0$, that is*

$$\|Rc\| - \|R\tilde{c}\| \leq \varepsilon\|F(x_0) - x_0\|,$$

*then*

$$\|(F - I)x_e\| \leq \left( \max_{x \in [0,\rho]} |p_*^C(x)| + 3C\alpha k + \varepsilon \right) \|(F - I)x_0\|.$$

*Under Assumption 2.2.2, this bound holds with $\alpha = kC\eta\|F(x_0) - x_0\|$.*

The result of Proposition 2.3.2 is independent of the dimension of the ambient space. Moreover, we can also get dimension dependent local superlinear convergence.

**Remark 2.3.4.** *Let $\rho \in ]0, 1[$ and $F$ be satisfying Assumption 2.2.2. Let $x_e \in \mathbb{R}^d$ be the output of Algorithm 2.2 initiated at $x_0 \in \mathbb{R}^d$ with $C \geq 1$ and $k \geq 1$. A slight modification in the proof of Proposition 2.3.2 yields*

$$\|F(x_e) - x_e\| \leq \left( \min_{\substack{p \in \mathbb{R}_k[X] \\ p(1)=1, \|p\|_1 \leq C}} \|p(G)\| + 3C^2\eta k^2 \|F(x_0) - x_0\| \right) \|F(x_0) - x_0\|.$$

*Let $\lambda_1, \ldots, \lambda_d \in [0, \rho[$ be the eigenvalues of $G$, when $C \geq \frac{(1+\rho)^n}{(1-\rho)^n}$ the polynomial $\chi(X) = \frac{\prod_{i=1}^d (X - \lambda_i)}{\prod_{i=1}^d (1 - \lambda_i)}$ satisfies $\chi(G) = 0$, $\chi(1) = 1$ and $\|\chi\|_1 \leq C$, thus for $k = d$*

$$\|F(x_e) - x_e\| \leq 3C^2\eta d^2 \|F(x_0) - x_0\|^2,$$

*which gives local superlinear convergence. Setting $k = d$ is of course somewhat impractical when the ambient dimension of the problem gets large.*

In the rest of the chapter, we focus on convergence rates that are dimension independent. Proposition 2.3.2 highlights a trade-off between (i) allowing coefficients to have larger magnitudes, i.e., via a large $C$, leading to a smaller $\max_{x \in [0,\rho]} |p_*^C(x)|$ that gets closer to the optimal rate $\rho_*$, and (ii) diminishing $C$ to better control the nonlinear part of $F$ but getting a slower rate $\max_{x \in [0,\rho]} |p_*^C(x)|$, closer to $\rho^k$. In the next section, we bound $\max_{x \in [0,\rho]} |p_*^C(x)|$ as a function of $C$, to make this trade-off explicit.

## 2.4 Constrained Chebyshev problem

We have seen in Section 2.1 that the Chebyshev problem, recalled in the following theorem, is central to many results of numerical analysis. For instance, it is used to provide convergence rates for several algorithms such as Lanczos' method for eigenvalue computations [Golub and Van Loan, 1990], conjugate gradients [Shewchuk, 1994], Anderson acceleration, or Chebyshev iterations [Golub and Varga, 1961, Nemirovskiy and Polyak, 1984, Nemirovsky, 1992].

We recall the fundamental result on rescaled Chebyshev polynomials from Golub and Varga [1961, Section 3] (which is a particular instance of Proposition 2.1.4).

**Theorem 2.4.1** (Golub and Varga [1961])**.** *Let $\rho \in ]0,1[$ and $k > 0$, we call Chebyshev problem of degree $k$ on $[0,\rho]$ the following optimization problem on polynomials*

$$\rho_* := \min_{\substack{p \in \mathbb{R}_k[X] \\ p(1)=1}} \max_{x \in [0,\rho]} |p(x)|, \tag{Cheb}$$

*whose solution is $p_*(X) = R_k^{[0,\rho]}(X)$, and $\rho_* = \frac{2\beta^k}{1+\beta^{2k}}$ with $\beta = \frac{1-\sqrt{1-\rho}}{1+\sqrt{1-\rho}}$.*

The following corollary extends the result of Theorem 2.4.1 and will be useful at the end of this section.

**Corollary 2.4.2.** *Let $\rho \in ]0,1[$, $k > 0$ and $\varepsilon \geq 0$. It holds that*

$$\rho_\varepsilon := \min_{\substack{p \in \mathbb{R}_k[X] \\ p(1)=1}} \max_{x \in [-\varepsilon,\rho]} |p(x)|, \tag{2.8}$$

*whose solution is the rescaled Chebyshev polynomial $R_k^{[-\varepsilon,\rho]}(X)$, and $\rho_\varepsilon = \frac{2\beta_\varepsilon^k}{1+\beta_\varepsilon^{2k}}$ with $\beta_\varepsilon = 1 - \sqrt{1 - \frac{\rho+\varepsilon}{1+\varepsilon}} \Big/ 1 + \sqrt{1 - \frac{\rho+\varepsilon}{1+\varepsilon}}$.*

*Proof.* This is an application of Proposition 2.1.4 with $a = -\varepsilon$ and $b = \rho$. ∎

We have seen in Proposition 2.3.2 that we need to control the optimal value of a slightly modified version of (Cheb) with an additional constraint on the $\ell_1$ norm of the vector of coefficients of the polynomial. Adding this constraint breaks the explicit result of Theorem 2.4.1 and no closed form solution for this constrained Chebyshev problem is known for arbitrary choices of $C$. In this section we seek upper bounds on the optimal value to this problem.

Let $k > 0$ and $\rho \in ]0,1[$, we are interested in the following constrained Chebyshev problem

$$\tilde{\rho}(C) := \min_{\substack{p \in \mathbb{R}_k[X] \\ p(1)=1 \\ \|p\|_1 \leq C}} \max_{x \in [0,\rho]} |p(x)|. \tag{Cstr-Cheb}$$

Before detailing explicit upper bounds on this problem, we first explain how to compute this $\tilde{\rho}(C)$ numerically for $C \geq 1$. Note that the feasible set is trivially empty when $C < 1$.

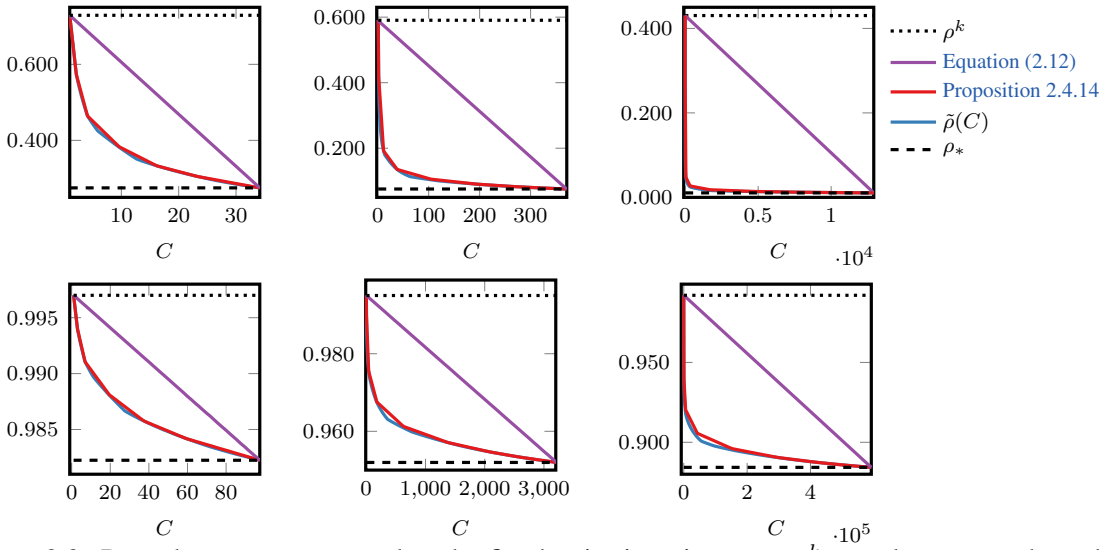


Figure 2.3: Dotted curves correspond to the fixed point iterations rate $\rho^k$, purple ones are bounds on $\tilde{\rho}(C)$ from (2.12) using convexity. Red curves correspond to the bound on $\tilde{\rho}(C)$ presented in Proposition 2.4.14 with $M = k$, blue ones correspond to numerical solutions to (2.10) (i.e., numerical value of $\tilde{\rho}(C)$) and the dashed ones to accelerated rate $\rho_*$ defined in (Cheb). On x-axis $C$ goes from 1 to $C_*$ defined in (2.11). Top : $\rho = 0.9$. Bottom: $\rho = 0.999$. Left: $k = 3$. Middle: $k = 5$. Right: $k = 8$.

### 2.4.1 Numerical solutions

When $C \geq 1$, the problem (Cstr-Cheb) has a non empty feasible set, and this feasible set is convex (intersection of an affine space with an $\ell_1$ ball). The objective function is a norm on $\mathbb{R}_k[X]$, hence is convex. The problem (Cstr-Cheb) is equivalent to

$$\begin{aligned} \tilde{\rho}(C) = \quad & \min \quad t \\ & p \in \mathbb{R}_k[X],\ t \in \mathbb{R}, \\ & p(1) = 1,\ \|p\|_1 \leq C, \\ & -t \leq p(x) \leq t,\ \forall x \in [0, \rho]. \end{aligned} \tag{2.9}$$

This problem involves polynomial positivity constraints on a bounded interval. A classical argument to transform this local positivity into positivity on $\mathbb{R}$ uses the following change of variable.

$$p(x) \geq 0\ \forall x \in [0, \rho] \iff (1 + x^2)^k p\left(\rho \tfrac{x^2}{1+x^2}\right) \geq 0\ \forall x \in \mathbb{R}.$$

Positivity constraints for univariate polynomials can be expressed using a sum of squares (SOS) formulation [Parrilo, 2000, Lasserre, 2001] (see e.g., Magron et al. [2019, Theorem 1] for a short proof). Standard packages can be used to solve the following reformulation of (2.9) with SOS constraints.

$$\begin{aligned} \tilde{\rho}(C) = \quad & \min \quad t \\ & p \in \mathbb{R}_k[X],\ t \in \mathbb{R}, \\ & p(1) = 1,\ \|p\|_1 \leq C, \\ & (1 + x^2)^k p(\rho \tfrac{x^2}{1+x^2}) + (1 + x^2)^k t \geq 0\ \forall x \in \mathbb{R}, \\ & (1 + x^2)^k t - (1 + x^2)^k p(\rho \tfrac{x^2}{1+x^2}) \geq 0\ \forall x \in \mathbb{R}. \end{aligned} \tag{2.10}$$

We used YALMIP [Löfberg, 2004] and MOSEK [Mosek, 2010] and numerical solutions to (2.10) are detailed in Figure 2.3 (in blue) for a few values of $\rho$ and $k$.

### 2.4.2 Exact and upper bounds

The main goal of this section is to provide an explicit upper bound for the function $\tilde{\rho}(C)$ defined in (Cstr-Cheb), which we later combine with the result of Proposition 2.3.2.

#### Naive upper bound and base properties

We start by presenting a property of the function $\tilde{\rho}$ that will be very useful to stitch together several upper bounds that will be derived on $\tilde{\rho}$ in what follows.

**Proposition 2.4.3.** *The function $\tilde{\rho}$ defined in* (Cstr-Cheb) *is convex on* $[1, +\infty[$.

*Proof.* Let $C_0, C_1 \in [1, +\infty[$ and $t \in [0, 1]$, when $p_0$ and $p_1$ are feasible points for problem (Cstr-Cheb) with $C$ equal to $C_0$ and $C_1$, then $(1-t)p_0 + tp_1$ is feasible for problem (Cstr-Cheb) with $C = (1-t)C_0 + tC_1$. Thus, by convexity of the objective function we have that $\tilde{\rho}((1-t)C_0 + tC_1) \leq (1-t)\tilde{\rho}(C_0) + t\tilde{\rho}(C_1)$. ■

We write $C_*$ the $\ell_1$-norm of the rescaled Chebyshev polynomial $p_* = R_k^{[0,\rho]}$ of Theorem 2.4.1, i.e.

$$C_* = \|R_k^{[0,\rho]}\|_1, \text{ where } R_k^{[0,\rho]} \text{ solves (Cheb).} \tag{2.11}$$

We start with a few observations on the behavior of $\tilde{\rho}$ at the boundaries of its domain.

**Remark 2.4.4.** *From Theorem 2.4.1, when $C$ is larger than $C_*$, problem* (Cstr-Cheb) *becomes unconstrained and $\tilde{\rho}(C)$ is constant equal to $\rho_*$.*

**Remark 2.4.5.** *When $C = 1$, the feasible set of* (Cstr-Cheb) *consists only of convex combinations of monomials of degree smaller than $k$. Among them, $X^k$ has the minimal absolute value on $[0, \rho]$ and $\tilde{\rho}(1) = \rho^k$.*

Based on the two previous remarks and Proposition 2.4.3 we obtain a first natural upper bound on $\tilde{\rho}$ written

$$\tilde{\rho}(C) \leq \tfrac{C_*-C}{C_*-1}\rho^k + \tfrac{C-1}{C_*-1}\rho_*, \qquad \text{for } C \in [1, C_*]. \tag{2.12}$$

This is a very coarse upper bound since $C_* \gg 1$. Indeed we can observe in Figure 2.3 that there is an important gap between $\tilde{\rho}$ and the coarse upper bound from (2.12) that is displayed in purple. Bellow, we show that using a refined set of sample points in $[1, C_*]$ along with convexity of $\tilde{\rho}$ allows obtaining more precise upper bounds.

#### Behavior for C close to 1

It turns out that when $C$ is close to 1, the behavior of $\tilde{\rho}(C)$ can be explicitly characterized. Indeed, in the next lemma we provide an explicit expression for $\tilde{\rho}(C)$ when $C$ is in an explicit neighborhood of 1.

**Lemma 2.4.6.** *Let $C_1 = \frac{2+\rho^k}{2-\rho^k}$, for $C \in [1, C_1]$ we have the following expression for $\tilde{\rho}$*

$$\tilde{\rho}(C) = \tfrac{C+1}{2}\rho^k - \tfrac{C-1}{2}.$$

*Proof.* Let us show that $p(X) = \frac{C+1}{2}X^k - \frac{C-1}{2}$ solves of (Cstr-Cheb). First notice that $p$ is feasible as $\|p\|_1 = C$ and $p(1) = 1$. In addition, since $p$ is increasing on $[0, \rho]$, $|p|$ reach its maximum on the boundary and $\max_{x\in[0,\rho]} |p(x)| = \max(|p(\rho)|, |p(0)|) = p(\rho) = \frac{C+1}{2}\rho^k - \frac{C-1}{2}$ using that $C \in [1, \frac{2+\rho^k}{2-\rho^k}]$.

Let $q$ be another feasible polynomial such that $q = \sum_{i=0}^k q_i X^i$, $\sum_{i=0}^k q_i = 1$ and $\sum_{i=0}^k |q_i| \leq C$. We show that $|q(\rho)| \geq |p(\rho)|$. First we have that

$$q(\rho) = \sum_{q_i\geq 0} q_i\rho^i + \sum_{q_i\leq 0} q_i\rho^i \geq \sum_{q_i\geq 0} q_i\rho^k + \sum_{q_i\leq 0} q_i = \sum_{q_i\geq 0} q_i\rho^k + \left(1 - \sum_{q_i\geq 0} q_i\right).$$

In addition, one notices that $\sum_{q_i\geq 0} q_i - \sum_{q_i\leq 0} q_i = \sum_{i=0}^k |q_i| \leq C$ thus using that $\sum_{q_i\leq 0} q_i = 1 - \sum_{q_i\geq 0} q_i$, we obtain $\sum_{q_i\geq 0} q_i \leq \frac{C+1}{2}$, and

$$q(\rho) \geq \tfrac{C+1}{2}(\rho^k - 1) + 1 = p(\rho) > 0.$$

Thus $|q(\rho)| = q(\rho) \geq p(\rho) = \max_{x\in[0,\rho]} |p(x)|$. Then $\max_{x\in[0,\rho]} |q(x)| \geq \max_{x\in[0,\rho]} |p(x)|$ so $p$ is an optimal solution of (Cstr-Cheb). ■

Using Remark 2.4.4 and Lemma 2.4.6, we can obtain exact expression for $\tilde{\rho}$ when $k = 1$.

**Remark 2.4.7.** *For $k = 1$, we have $C_* = \frac{2+\rho}{2-\rho} = C_1$ and*

$$\tilde{\rho}(C) = \begin{cases} \frac{C+1}{2}\rho^k - \frac{C-1}{2} & \text{for } C \in [1, C_1] \\ \frac{2}{2-\rho} & \text{for } C \geq C_1 \end{cases}.$$

We can also give an explicit form for solutions of (Cstr-Cheb) in a neighborhood of $C_*$ as detailed below.

### Behavior for C around $C_*$

In this section, we show that solutions to the Chebyshev problem with light constraints ($C$ close to $C_*$) are also rescaled Chebyshev polynomials (see Definition 2.1.3) on a segment $[-\varepsilon, \rho]$ instead of $[0, \rho]$, with $\varepsilon \geq 0$.

**Theorem 2.4.8.** *Let $\rho \in ]0, 1[$, $k > 0$ and*

$$\tilde{\varepsilon} = \rho\frac{1+\cos(\frac{2k-1}{2k}\pi)}{1-\cos(\frac{2k-1}{2k}\pi)}.$$

*For any $\varepsilon \in [0, \tilde{\varepsilon}]$ we have*

$$R_k^{[-\varepsilon,\rho]} \in \underset{\substack{p\in\mathbb{R}_k[X],\, p(1)=1 \\ \|p\|_1\leq \left\|R_k^{[-\varepsilon,\rho]}\right\|_1}}{\operatorname{argmin}} \max_{x\in[0,\rho]} |p(x)|,$$

*which implies*

$$\tilde{\rho}\left(\left\|R_k^{[-\varepsilon,\rho]}\right\|_1\right) = \max_{x\in[-\varepsilon,\rho]} \left|R_k^{[-\varepsilon,\rho]}(x)\right|.$$

In order to prove Theorem 2.4.8, we use some intermediary results on rescaled Chebyshev polynomials, listed on the following Lemmas.

First we provide results on the equioscillation properties of rescaled Chebyshev polynomials $R_k^{[-\varepsilon,\rho]}$ when $\varepsilon$ is small enough.

**Lemma 2.4.9.** *Let $k > 0$ and $\rho \in [0, 1[$. For $\varepsilon \in [0, \tilde{\varepsilon}]$ with $\tilde{\varepsilon} = \rho\frac{1+\cos(\frac{2k-1}{2k}\pi)}{1-\cos(\frac{2k-1}{2k}\pi)}$ we have the following properties of $R_k^{[-\varepsilon,\rho]} = \frac{T_k(\frac{2(X+\varepsilon)}{\rho+\varepsilon}-1)}{T_k(\frac{2(1+\varepsilon)}{\rho+\varepsilon}-1)}$.*

*(i)* $\left|R_k^{[-\varepsilon,\rho]}(X)\right|$ *is maximal on the* $m_i = \frac{(\rho+\varepsilon)\cos(\frac{i\pi}{k})+\rho-\varepsilon}{2} \in [-\varepsilon, \rho]$ *for* $i = 0, \dots, k$ *and* $\operatorname{sign}(R_k^{[-\varepsilon,\rho]}(m_i)) = (-1)^i$.

*(ii) Let* $c \in \mathbb{R}^{k+1}$ *such that* $R_k^{[-\varepsilon,\rho]}(X) = \sum_{i=0}^k c_i X^i$. *Then* $\operatorname{sign}(c_i) = (-1)^{k-i}$ *for* $i = 1, \dots, k$ *and* $(-1)^k c_0 \geq 0$.

*Proof.* The Chebyshev polynomial of first kind $T_k(X)$ is defined such that

$$T_k(\cos(\theta)) = \cos(k\theta) \text{ for all } \theta \in \mathbb{R}.$$

Using this property, *(i)* is obtained by observing that $\max_{x\in[-1,1]} |T_k(x)| = 1$ is attained for $x_i = \cos(\frac{i\pi}{k})$ with $i = 0, \dots, k$. In particular $T_k(x_i) = (-1)^i$. Thus $|R_k^{[-\varepsilon,\rho]}|$ has its maxima on

$$m_i = \frac{(\rho+\varepsilon)\cos(\frac{i\pi}{k})+\rho-\varepsilon}{2}, \quad i = 0, \dots, k$$

and $R_k^{[-\varepsilon,\rho]}(m_i) = (-1)^i$.

Let us now prove *(ii)*. From the definition of $T_k$, we get that the roots of $T_k$ are the $(z_i)_{i\in[0,k-1]} = (\cos\left(\frac{2i+1}{2k}\pi\right))_{i\in[0,k-1]} \in [-1, 1]$. The roots of $R_k^{[-\varepsilon,\rho]}(X)$ are the $z_i^\varepsilon$ defined such that $\frac{2z_i^\varepsilon}{\rho+\varepsilon} - \frac{\rho-\varepsilon}{\rho+\varepsilon} = z_i$. This corresponds to

$$z_i^\varepsilon = \frac{(\rho+\varepsilon)z_i+\rho-\varepsilon}{2} \in [-\varepsilon, \rho], \quad i = 0, \dots, k-1.$$

The smallest root $z_{k-1}^\varepsilon = \frac{(\rho+\varepsilon)\cos(\frac{2k-1}{2k}\pi)+\rho-\varepsilon}{2}$ is nonnegative for $\varepsilon \in [0, \rho\frac{1+\cos(\frac{2k-1}{2k}\pi)}{1-\cos(\frac{2k-1}{2k}\pi)}] = [0, \tilde{\varepsilon}]$. This means that this for choice of $\varepsilon$, all the roots of $R_k^{[-\varepsilon,\rho]}$ are in $[0, \rho]$.

One can thus express $R_k^{[-\varepsilon,\rho]}$ using its roots as $R_k^{[-\varepsilon,\rho]}(x) = a\prod_{i=0}^{k-1}(x - z_i^\varepsilon)$ with $a$ the leading coefficients. Using that the leading coefficient of $T_k$ is $2^{k-1}$ and that $T_k(\frac{2(1+\varepsilon)}{\rho+\varepsilon} - 1) > 0$ since $\frac{2(1+\varepsilon)}{\rho+\varepsilon} - 1 > 1$, we have $a > 0$. By developing the product we have

$$R_k^{[-\varepsilon,\rho]}(x) = a\left(\sum_{j=1}^{k}(-1)^{k-j}x^j \sum_{0<i_0<\dots<i_{k-j}} z_{i_0}^\varepsilon \cdots z_{i_{k-j}}^\varepsilon + x^k\right),$$

which gives us (ii). ∎

Then we study properties of some polynomial descent directions $h$ on the maximal absolute value on $[0, \rho]$ at the point $R_k^{[-\varepsilon,\rho]}$.

**Lemma 2.4.10.** *Let $k > 0$ and $0 \leq \varepsilon \leq \tilde{\varepsilon}$, suppose that there exists a nonzero polynomial $h \in \mathbb{R}_k[X]$ satisfying*

$$\begin{aligned}
&(i) \quad h(1) = 0.\\
&(ii) \quad \max_{x\in[0,\rho]} |h(x)| \leq \tfrac{1}{2} \max_{x\in[0,\rho]} |R_k^{[-\varepsilon,\rho]}(x)|.\\
&(iii) \quad \max_{x\in[0,\rho]} |R_k^{[-\varepsilon,\rho]}(x) + h(x)| < \max_{x\in[0,\rho]} |R_k^{[-\varepsilon,\rho]}(x)|.
\end{aligned}$$

*Then, $h$ possesses $k$ distinct roots in $]0,1]$, $(-1)^k h(-1) > 0$ and $(-1)^k h(0) > 0$.*

*Proof.* From Lemma 2.4.9 we know that $|R_k^{[-\varepsilon,\rho]}|$ is maximal on $[-\varepsilon,\rho]$ at the $m_i = \frac{(\rho+\varepsilon)\cos(\frac{i\pi}{k})+\rho-\varepsilon}{2}$ for $i = 0,\dots,k$ and $\operatorname{sign}(R_k^{[-\varepsilon,\rho]}(m_i)) = (-1)^i$. In addition, $m_i \in ]0,\rho]$ for $i = 0,\dots,k-1$. Indeed the $m_i$ are in decreasing order and $m_{k-1} = \frac{(\rho+\varepsilon)\cos(\frac{(k-1)\pi}{k})+\rho-\varepsilon}{2} > \frac{(\rho+\varepsilon)\cos(\frac{(2k-1)\pi}{2k})+\rho-\varepsilon}{2} \geq \frac{(\rho+\tilde{\varepsilon})\cos(\frac{(2k-1)\pi}{2k})+\rho-\tilde{\varepsilon}}{2} = 0$ since $\varepsilon \in [0,\tilde{\varepsilon}]$.

It follows from (ii) that $|h(m_i)| \leq \frac{1}{2}|R_k^{[-\varepsilon,\rho]}(m_i)|$ for $i = 0,\dots,k-1$ which implies that $|R_k^{[-\varepsilon,\rho]}(m_i) + h(m_i)| = |R_k^{[-\varepsilon,\rho]}(m_i)| + \operatorname{sign}(R_k^{[-\varepsilon,\rho]}(m_i))h(m_i) = |R_k^{[-\varepsilon,\rho]}(m_i)| + (-1)^i h(m_i)$. Together with (iii) this leads to $(-1)^i h(m_i) < 0$ for $i = 0,\dots,k-1$.

Because $h$ alternates sign between $m_i$'s, the mean value theorem implies that $h$ possesses a root inside each interval $]m_{i+1}, m_i[ \subset ]0,1[$ for $i = 0,\dots,k-2$. Along with (i), this shows that $h$ has $k$ distinct roots in $]m_{k-1},1] \subset ]0,1]$. Furthermore, $h$ keeps the same sign on $]-\infty, m_{k-1}]$ (since it already has $k$ roots) which is $(-1)^k$. In particular, it implies that $(-1)^k h(-1) > 0$ and $(-1)^k h(0) > 0$ reaching the desired statements. ∎

We can finally prove Theorem 2.4.8:

*Proof.* We proceed by contradiction: we assume that $R_k^{[-\varepsilon,\rho]}$ is not a solution to the constrained Chebyshev problem and show that it leads to a contradiction.

Assume that $R_k^{[-\varepsilon,\rho]}$ is not a global minimum of (Cstr-Cheb), $R_k^{[-\varepsilon,\rho]}$ is not a local minimum either, therefore for all $\delta > 0$, there exists a nonzero polynomial $h \in \mathbb{R}_k[X]$ such that

$$\begin{aligned}
&(i) \quad R_k^{[-\varepsilon,\rho]}(1) + h(1) = 1. && \text{(feasibility)}\\
&(ii) \quad \left\| R_k^{[-\varepsilon,\rho]} + h \right\|_1 \leq \left\| R_k^{[-\varepsilon,\rho]} \right\|_1. && \text{(feasibility)}\\
&(iii) \quad \max_{x\in[0,\rho]} |h(x)| \leq \delta. && \text{(not a local minimum)}\\
&(iv) \quad \max_{x\in[0,\rho]} |R_k^{[-\varepsilon,\rho]}(x) + h(x)| < \max_{x\in[0,\rho]} |R_k^{[-\varepsilon,\rho]}(x)|. && \text{(not a minimum).}
\end{aligned}$$

For $\delta < \frac{1}{2}\max_{x\in[0,\rho]} |R_k^{[-\varepsilon,\rho]}(x)|$, (i), (iii) and (iv) correspond to the assumptions of Lemma 2.4.10 for $h$. This implies that it possesses $k$ roots in $]0,1[$ , $(-1)^k h(-1) > 0$ and $(-1)^k h(0) > 0$.

Then, writing $R_k^{[-\varepsilon,\rho]}(x) = \sum_{i=0}^k c_i x^i$ and $h(x) = \sum_{i=0}^k h_i x^i$, Lemma 2.4.9 allows concluding that

$$\begin{aligned}
&(v) \quad c_i \neq 0 \text{ and } \operatorname{sign}(c_i) = (-1)^{k+i} \text{ for } i = 1,\dots,k\\
&(vi) \quad (-1)^k c_0 \geq 0.
\end{aligned}$$

From (vi) and the fact that $(-1)^k h(0) > 0$, we have $|c_0 + h_0| = (-1)^k(c_0 + h_0) = |c_0| + (-1)^k h_0$. Hence, for $\delta$ small enough, it follows that $0 < \max_{i=1,\dots,k} |h_i| < \min_{i=1,\dots,k} |c_i|$ and we obtain

$$|c_i + h_i| = |c_i| + \operatorname{sign}(c_i)h_i = |c_i| + (-1)^{k+i} h_i,$$

where the second equality follows from (v).

It remains to express the $\ell_1$ norm of $R_k^{[-\varepsilon,\rho]} + h$ as

$$\begin{aligned}\left\|R_k^{[-\varepsilon,\rho]} + h\right\|_1 &= \sum_{i=0}^{k} |c_i + h_i| = \sum_{i=0}^{k} |c_i| + (-1)^{k+i} h_i \\ &= \left\|R_k^{[-\varepsilon,\rho]}\right\|_1 + (-1)^k h(-1).\end{aligned}$$

Combining (ii) with the previous equality leads to $(-1)^k h(-1) \leq 0$ which is in contradiction with $(-1)^k h(-1)$ obtained earlier.

Therefore, $R_k^{[-\varepsilon,\rho]}$ has to be a solution of (Cstr-Cheb), reaching the desired claim. ■

As mentioned in this proof, the coefficients of $R_k^{[-\varepsilon,\rho]}$ for $\varepsilon \in [0, \tilde{\varepsilon}]$ have alternating signs, so $\left\|R_k^{[-\varepsilon,\rho]}\right\|_1$ is in fact $|R_k^{[-\varepsilon,\rho]}(-1)|$. This relation is key in the proof of the previous theorem, and is the main motivation behind the choice of the $\ell_1$ norm on coefficients (versus e.g. $\ell_2$). Furthermore, this yields a somewhat simple expression for $\left\|R_k^{[-\varepsilon,\rho]}\right\|_1$, as follows.

**Lemma 2.4.11.** *Let $\rho \in ]0,1[$, and $k > 0$. For any $\varepsilon \in [0, \tilde{\varepsilon}]$ with*

$$\tilde{\varepsilon} = \rho \frac{1+\cos(\frac{2k-1}{2k}\pi)}{1-\cos(\frac{2k-1}{2k}\pi)},$$

*we have*

$$\|R_k^{[-\varepsilon,\rho]}\|_1 = \frac{\left(1+\frac{\rho-\varepsilon}{2}-\sqrt{(1+\rho)(1-\varepsilon)}\right)^k + \left(1+\frac{\rho-\varepsilon}{2}+\sqrt{(1+\rho)(1-\varepsilon)}\right)^k}{\left(1+\frac{\rho-\varepsilon}{2}-\sqrt{(1-\rho)(1+\varepsilon)}\right)^k + \left(1+\frac{\rho-\varepsilon}{2}+\sqrt{(1-\rho)(1+\varepsilon)}\right)^k}.$$

*Furthermore, the function $\varepsilon \to \left\|R_k^{[-\varepsilon,\rho]}\right\|_1$ is continuous and decreasing on $[0, \tilde{\varepsilon}]$.*

*Proof.* Using $\|R_k^{[-\varepsilon,\rho]}\|_1 = |R_k^{[-\varepsilon,\rho]}(-1)|$, we apply the classical expression for $T_k(x)$ with $|x| \geq 1$ (see e.g., Mason and Handscomb [2002, Eq 1.49]) $T_k(x) = \frac{1}{2}\left((x-\sqrt{x^2-1})^k + (x+\sqrt{x^2-1})^k\right)$. Using this formula, we arrive to (after a bit of work)

$$\left\|R_k^{[-\varepsilon,\rho]}\right\|_1 = \frac{\left|T_k(\frac{-2-\rho+\varepsilon}{\rho+\varepsilon})\right|}{\left|T_k(\frac{2-\rho+\varepsilon}{\rho+\varepsilon})\right|} = \frac{\left(1+\frac{\rho-\varepsilon}{2}-\sqrt{(1+\rho)(1-\varepsilon)}\right)^k + \left(1+\frac{\rho-\varepsilon}{2}+\sqrt{(1+\rho)(1-\varepsilon)}\right)^k}{\left(1-\frac{\rho-\varepsilon}{2}-\sqrt{(1-\rho)(1+\varepsilon)}\right)^k + \left(1-\frac{\rho-\varepsilon}{2}+\sqrt{(1-\rho)(1+\varepsilon)}\right)^k}.$$

A base study of variations reveals that the numerator is decreasing and the denominator increasing on $[0, \tilde{\varepsilon}]$. ■

**Remark 2.4.12.** *In particular, one can express the value of $C_*$ using Lemma 2.4.11 applied to the unconstrained Chebyshev problem* (Cheb) *($\varepsilon = 0$), yielding*

$$C_* = \frac{(2+\rho-2\sqrt{1+\rho})^k + (2+\rho+2\sqrt{1+\rho})^k}{(2-\rho-2\sqrt{1-\rho})^k + (2-\rho+2\sqrt{1-\rho})^k}.$$

**Remark 2.4.13.** *Theorem 2.4.8 and Lemma 2.4.11 do not provide explicit expressions of $\tilde{\rho}(C)$ for $C \in [\tilde{C}, C_*]$ with $\tilde{C} = \|R_k^{[\tilde{\varepsilon},\rho]}\|_1$. Indeed, we cannot explicitly invert the relation $\varepsilon \to \|R_k^{[-\varepsilon,\rho]}\|_1$. However one can get arbitrarily tight upper bounds by sampling $(\varepsilon_i)_{i\in[1,M]} \in [0, \tilde{\varepsilon}]$. Then, one can compute $C_i = \|R_k^{[\varepsilon_i,\rho]}\|_1$ explicitly using Lemma 2.4.11. Note that since $\varepsilon \to \left\|R_k^{[-\varepsilon,\rho]}\right\|_1$ is continuous and decreasing on $[0, \tilde{\varepsilon}]$, we can obtain an arbitrarily good covering of $[\tilde{C}, C_*]$ using the $C_i$. Finally, using convexity from Proposition 2.4.3 to interpolate linearly between the $C_i$ and $\tilde{\rho}(C_i)$, provides a piecewise linear upper bound on $[\tilde{C}, C_*]$ which can be made arbitrarily close to $\tilde{\rho}$ by increasing $M$.*

Note however that the interval $[\tilde{C}, C_*]$ is actually quite narrow compared with $[1, C_*]$. We describe the construction of upper bounds for all $C \geq 1$ in the next section.

### Construction of upper bounds for all values of C

To construct an upper bounds on $\tilde{\rho}(C)$ for all $C \in [1, C_*]$, we use the idea presented above, based on bounding $\tilde{\rho}$ at a finite number of points, then using convexity to interpolate upper bounds between these points. Given $M \in \mathbb{N}^*$ accounting for the number of intermediate breaking points, the upper bound is built as follows.

(i) Select $M+2$ constraint parameters $C_i \in [1, C_*]$ for $i = 0, \dots, M+1$ with $C_0 = 1$, $C_1 = \frac{2+\rho^k}{2-\rho^k}$ (from Lemma 2.4.6) and $C_{M+1} = C_*$.

(ii) Using feasible polynomials of (Cstr-Cheb), obtain $\rho_i$ such that $\tilde{\rho}(C_i) \leq \rho_i$ for $i = 0, \dots, M+1$, with $\rho_0 = \rho^k$, $\rho_1 = \tilde{\rho}(C_1) = \frac{\rho^k}{2-\rho^k}$ (from Lemma 2.4.6) and $\rho_{M+1} = \rho_*$.

(iii) Use the lower convex hull of the $(C_i, \rho_i)$ as an upper bound on $\tilde{\rho}$ on $[1, C_*]$.

Note that we only focus on $[1, C_*]$ since we know that $\tilde{\rho}(C) = \rho_*$ when $C \geq C_*$ (see Remark 2.4.12).

**Proposition 2.4.14.** *Let $k > 2$ , $\rho \in ]0,1[$ and $M \geq 2$, $(\varepsilon_i)_{i\in[2,M]} = \left(\frac{\rho}{2^{i-2}}\right)_{i\in[2,M]}$. Let*

$$(C_i)_{i\in[2,M]} = \left(\min \left\|R_k^{[\varepsilon_i,\rho]}\right\|_1, C_*\right)_{i\in[2,M]}, \quad C_0 = 1, \quad C_1 = \tfrac{2+\rho^k}{2-\rho^k}, \quad \text{and } C_{M+1} = C_*.$$

*We denote by $(\rho_i)_{i\in[2,M]} = \left(\frac{2\beta_i^k}{1+\beta_i^{2k}}\right)_{i\in[2,M]}$, with $\beta_i = 1 - \sqrt{1-\frac{\rho+\varepsilon_i}{1+\varepsilon_i}} \Big/ 1 + \sqrt{1-\frac{\rho+\varepsilon_i}{1+\varepsilon_i}}$, $\rho_0 = \rho^k$, $\rho_1 = \frac{\rho^k}{2-\rho^k}$ and $\rho_{M+1} = \rho_*$. Then, we index $C_{[i]}$ such that $1 = C_0 = C_{[0]} \leq C_{[1]} \leq \ldots \leq C_{[M+1]} = C_{M+1} = C_*$, and define $\tilde{\rho}_b$ on $[1, +\infty]$ as*

$$\tilde{\rho}_b(C) = \begin{cases} \min\limits_{\substack{j,l \\ C_{[j]} \leq C_{[i]} \\ C_{[l]} \geq C_{[i+1]}}} \frac{C-C_{[j]}}{C_{[l]}-C_{[j]}}\rho_{[l]} + \frac{C_{[l]}-C}{C_{[l]}-C_{[j]}}\rho_{[j]} & \text{for } C \in [C_{[i]}, C_{[i+1]}] \\ \rho_*, & \text{for } C \geq C_* \end{cases},$$

*which is an upper bound on $\tilde{\rho}$.*

*Proof.* The values $C_i = \left\|R_k^{[\varepsilon_i,\rho]}\right\|_1$ can computed explicitly using e.g. Mason and Handscomb [2002, Equation 2.18] to obtain the coefficients of $T_k$. If $C > C_*$ we saw that $\tilde{\rho}(C) = \rho_*$. Otherwise $C$ is between a $C_{[i]}$ and a $C_{[i+1]}$ and the result follows from the convexity of $\tilde{\rho}$. ∎

To select the $C_i$ in step (i), we rely on the intuition provided by Theorem 2.4.8 and on numerical observations. Indeed we noticed that for a large range of $\varepsilon$ (more precisely for $\varepsilon \in [0, \rho]$), $\max_{[0,\rho]} \left|R_k^{[-\varepsilon,\rho]}(x)\right|$ is a good upper bound for $\tilde{\rho}\left(\left\|R_k^{[-\varepsilon,\rho]}\right\|_1\right)$. Therefore, we sample $M-1$ values $\varepsilon_i \in [0,\rho]$ and use $C_i = \left\|R_k^{[\varepsilon_i,\rho]}\right\|_1$ in step (i). Then, we set $\rho_i = \max_{[0,\rho]} \left|R_k^{[\varepsilon_i,\rho]}(x)\right|$ ($= \rho_{\varepsilon_i}$ from Corollary 2.4.2) in step (ii) and finally apply step (iii) to get the upper bound. As shown in Figure 2.3, Proposition 2.4.14 provides upper bounds, represented in red on the figure, that are close to $\tilde{\rho}(C)$.

Setting $\varepsilon_2 = \rho$ in the previous proposition is motivated by two observations, (i) numerically $\rho_\varepsilon = \max_{x\in[0,\rho]} |R_k^{[\varepsilon,\rho]}(x)|$ is close to $\rho\left(\left\|R_k^{[-\varepsilon,\rho]}\right\|_1\right)$ for $\varepsilon \in [0,\rho]$ and diverges from it for larger $\varepsilon$, (ii) we can study $R_k^{[\rho,\rho]}$ and get a relatively simple expression for $\|R_k^{[\rho,\rho]}\|_1$ as described below.

**Lemma 2.4.15.** *Let $\rho \in ]0,1[$ and $k \geq 1$,*

$$C_2 := \left\|R_k^{[\rho,\rho]}\right\|_1 = \frac{(1-\sqrt{1+\rho^2})^k+(1+\sqrt{1+\rho^2})^k}{(1-\sqrt{1-\rho^2})^k+(1+\sqrt{1-\rho^2})^k}. \tag{2.13}$$

*Proof.* From Definition 2.1.3 we have $R_k^{[\rho,\rho]}(X) = \frac{T_k(\frac{X}{\rho})}{T_k(\frac{1}{\rho})}$. Then, noticing that $\|R_k^{[\rho,\rho]}\|_1 = \frac{\left|T_k(\frac{i}{\rho})\right|}{T_k(\frac{1}{\rho})}$ (with $i$ the unit imaginary number) allows to use the nice formulation for the value of Chebyshev polynomials (see e.g., Mason and Handscomb [2002, Eq 1.49])

$$\|R_k^{[\rho,\rho]}\|_1 = \frac{\left|\left(i-(i^2-\rho)^{1/2}\right)^k+\left(i-(i^2-\rho)^{1/2}\right)^k)\right|}{(1-\sqrt{1-\rho^2})^k+(1+\sqrt{1-\rho^2})^k} = C_2.$$

■

In order to get more insights on how this upper bound behaves at $C_2$, we study the regime $\rho \sim 1$.

**Remark 2.4.16.** *We have*

$$\rho_2 := \max_{x\in[0,\rho]} |R_k^{[\rho,\rho]}(x)| = \frac{2\rho^k}{(1+\sqrt{1-\rho^2})^k+(1-\sqrt{1-\rho^2})^k} \leq \rho^k.$$

*When $\rho \to 1$ we can show*

$$1-\rho^k \sim k(1-\rho),\ 1-\rho_* \sim k^2(1-\rho) \text{ and } 1-\rho_2 \sim \tfrac{k}{2k-1}k^2(1-\rho).$$

*In addition,*

$$C_2 \sim \frac{(1+\sqrt{2})^k+(1-\sqrt{2})^k}{(1+\sqrt{2})^{2k}+(1-\sqrt{2})^{2k}} C_* \leq \frac{1}{2^k} C_*.$$

*In the bad conditioned regime where $\rho \sim 1$, decreasing the constraint $C$ by a factor $2^k$, only deteriorates the convergence rate by a factor $\frac{k}{2k-1}$.*

We now, present a simpler and more practical upper bound, which corresponds to a scenario where the $C_i$'s are ordered.

**Remark 2.4.17.** *Following the notations of Proposition 2.4.14, a simpler upper bound when the $C_i$'s are ordered is*

$$\tilde{\rho}_{bo}(C) := \begin{cases} \frac{C-C_i}{C_{i+1}-C_i}\rho_{i+1} + \frac{C_{i+1}-C}{C_{i+1}-C_i}\rho_i & \text{for } C \in [C_i, C_{i+1}] \text{ and } 0 \leq i \leq M \\ \rho_* & \text{for } C \geq C_* \end{cases}. \tag{2.14}$$

*Numerically* $\left\|R_k^{[-\varepsilon,\rho]}\right\|_1$ *appears to be decreasing with* $\varepsilon$*, as the intuition suggests. Indeed, when* $\varepsilon$ *gets larger, the graph of* $R_k^{[-\varepsilon,\rho]}$ *exhibits wider oscillations, which would imply a decrease in the magnitude of its coefficients. For now, this remains a conjecture as we could not prove it formally. Note however that for* $M = 2$*, we can show (see Lemma 2.4.18) that* $C_0 < C_1 < C_2 < C_3$ *and thus, (2.14) defines a simplified upper bound.*

**Lemma 2.4.18.** *Let* $k \in \mathbb{N}$*,* $\rho < 1$*,* $C_*$ *is defined in (2.11) (an explicit value is provided in Remark 2.4.12) and* $C_2$ *in (2.13). It holds that*

$$C_1 = \tfrac{2+\rho^k}{2-\rho^k} \leq C_2 \text{ for } k > 1 \text{ and } C_2 \leq C_* \text{ for } k \geq 1.$$

*Proof.* We start from the expression of $C_2$

$$C_2 = \frac{(1-\sqrt{1+\rho^2})^k+(1+\sqrt{1+\rho^2})^k}{(1-\sqrt{1-\rho^2})^k+(1+\sqrt{1-\rho^2})^k} = \frac{\sum_{i=0}^{\lfloor k/2\rfloor}\binom{k}{2i}(1+\rho^2)^i}{\sum_{i=0}^{\lfloor k/2\rfloor}\binom{k}{2i}(1-\rho^2)^i}.$$

For obtaining $C_2 \geq \frac{2+\rho^k}{2-\rho^k}$ we need to show

$$(2-\rho^k)\sum_{i=0}^{\lfloor k/2\rfloor}\binom{k}{2i}(1+\rho^2)^i - (2+\rho^k)\sum_{i=0}^{\lfloor k/2\rfloor}\binom{k}{2i}(1-\rho^2)^i \geq 0,$$

and in particular we study

$$(2-\rho^k)(1+\rho^2)^i - (2+\rho^k)(1-\rho^2)^i.$$

When $i = 0$, this is equal to $-2\rho^k$, and when $i = 1$ this is equal to $4\rho^2 - 2\rho^k$. In addition, one can easily observe that it is an increasing function of $i$. Hence, it is nonnegative when $i \geq 1$. For $k \geq 2$, we can further write

$$\begin{aligned}(2-\rho^k)\sum_{i=0}^{\lfloor k/2\rfloor}\binom{k}{2i}(1+\rho^2)^i - (2+\rho^k)\sum_{i=0}^{\lfloor k/2\rfloor}\binom{k}{2i}(1-\rho^2)^i &\geq -2\rho^k + \binom{k}{2}(-2\rho^k+4\rho^2)\\ &\geq 4\rho^2(1-\rho^{k-2})\\ &\geq 0 \text{ strict inequality when } k > 2,\end{aligned}$$

and then

$$\boxed{C_2 \geq \tfrac{2+\rho^k}{2-\rho^k}} \text{ with strict inequality when } k > 2.$$

Finally, we show the second inequality between $C_*$ and $C_2$.

$$C_* = \frac{\left(2+\rho-2\sqrt{1+\rho}\right)^k+\left(2+\rho+2\sqrt{1+\rho}\right)^k}{\left(2-\rho-2\sqrt{1-\rho}\right)^k+\left(2-\rho+2\sqrt{1-\rho}\right)^k} = \frac{\sum_{i=0}^{\lfloor k/2\rfloor}\binom{k}{2i}(1+\frac{\rho}{2})^{k-2i}(1+\rho)^i}{\sum_{i=0}^{\lfloor k/2\rfloor}\binom{k}{2i}(1-\frac{\rho}{2})^{k-2i}(1-\rho)^i}.$$

When $k \geq 1$, $(1+\frac{\rho}{2})^{k-2i}(1+\rho)^i > (1+\rho^2)^i$ and $(1-\frac{\rho}{2})^{k-2i}(1-\rho)^i < (1-\rho^2)^i$ for $i \in [0, \lfloor\frac{k}{2}\rfloor]$ and thus

$$\boxed{C_* > C_2} \text{ when } k \geq 1,$$

reaching the desired conclusion. ∎

In the next section we use the simplified bound on the constrained Chebyshev problem of Remark 2.4.17 to provide explicit bounds on constrained Anderson acceleration for gradient step operators.

## 2.5 Convergence of CAA on gradient steps

As discussed above, combining Proposition 2.4.14 and Proposition 2.3.2 gives an explicit linear rate of convergence for one pass of Algorithm 2.2. In what follows, we focus on applications of these results to the optimization setting where $F$ is an operator representing an optimization method.

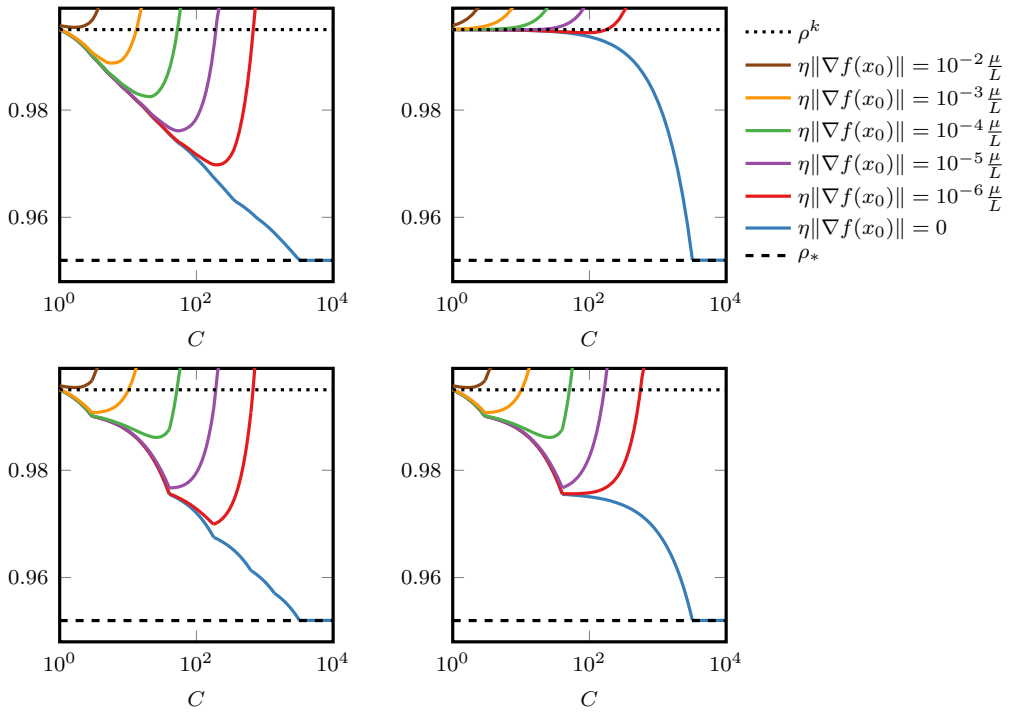


Figure 2.4: Bounds on the convergence rate of Algorithm 2.2 with $k = 5$, $\mu = 10^{-3}$ and $L = 1$. Top Left: bound from (2.15), Top Right: bound from (2.12), Bottom Left: bound from (2.16) with $M = k + 1$. Bottom Right: bound from (2.16) with $M = 2$. Note that the apparent nonconvexity of the bounds is due to the $x$-axis being represented in logarithmic scale.

### 2.5.1 Convergence bounds

In this section, we come back to the problem of accelerating convergence of a first-order method, and consider that $F$ is specifically encoding a gradient step of a function $f$ (see e.g., Scieur et al. [2020]). It is well known (see for instance Ryu and Boyd [2016]) that when $f$ is $\mu$-strongly convex with $L$-Lipschitz gradient for $0 < \mu < L$, $F = (I - \frac{1}{L}\nabla f)$ is a $\rho = (1 - \frac{\mu}{L})$-Lipschitz operator. In addition, we assume that $\nabla^2 f$, the Hessian of $f$, is $\eta$-Lipschitz for $\eta > 0$, which implies that $F$ satisfies Assumption 2.2.2.

Given $x_0 \in \mathbb{R}^d$, Proposition 2.3.2 shows that the output $x_e$ of Algorithm 2.2 with $k \geq 1$ and $C \geq 1$ satisfies

$$\|\nabla f(x_e)\| \leq \left(\tilde{\rho}(C) + 3\tfrac{\eta}{L}k^2C^2\|\nabla f(x_0)\|\right)\|\nabla f(x_0)\|, \tag{2.15}$$

where $\tilde{\rho}$ is defined in (Cstr-Cheb). When $C$ is fixed, there are two ways of for improving the convergence rate of CAA : (i) having a Hessian with a small Lipschitz constant $\eta$, which means being globally close to a quadratic, or (ii) being sufficiently close to the optimum (i.e., $\|\nabla f(x_0)\|$ small).

To make our bounds more concrete, we now combine (2.15) with the upper bound from Proposition 2.4.14. For clarity, we only consider the simple upper bound from Remark 2.4.17. The next proposition provides a range of values of $C$, depending on the perturbation parameter $\frac{\eta}{L^2}\|\nabla f(x_0)\|$ (which measures deviation from quadratic case), for which acceleration is guaranteed with Algorithm 2.2 compared to the baseline convergence rate $\rho^k$ after $k$ iterations of the fixed-step gradient method.

**Proposition 2.5.1.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a $\mu$-strongly convex function with $L$-Lipschitz gradient and $\eta$-Lipschitz Hessian. Let $x_e$ be the output of Algorithm 2.2 with $x_0 \in \mathbb{R}^d$, $k > 2$ and $C \geq 1$.*

$$\|\nabla f(x_e)\| \leq \left(\tilde{\rho}_{bo}(C) + 3\tfrac{\eta}{L^2}k^2C^2\|\nabla f(x_0)\|\right)\|\nabla f(x_0)\|. \tag{2.16}$$

*where $\tilde{\rho}_{bo}$ is defined in (2.14) with $M \geq 2$ and $\rho = 1 - \frac{\mu}{L}$. In addition,*

*(i) If $\frac{\eta}{L^2}\|\nabla f(x_0)\| < \frac{\rho^k(1-\rho^k)(2-\rho^k)}{3k^2(2+\rho^k)^2}$ then*

$$\exists\, \delta > 0 \text{ s.t. } \|\nabla f(x_e)\| < \rho^k \|\nabla f(x_0)\| \text{ for } C \in [\tfrac{2+\rho^k}{2-\rho^k} - \delta, \tfrac{2+\rho^k}{2-\rho^k} + \delta].$$

*(ii) If $\frac{\eta}{L^2}\|\nabla f(x_0)\| < \min(\frac{\rho^k(1-\rho^k)(2-\rho^k)}{3k^2(2+\rho^k)^2}, \frac{\rho^k-\rho_2}{3k^2C_2^2})$ then*

$$\|\nabla f(x_e)\| < \rho^k \|\nabla f(x_0)\| \text{ for } C \in [\tfrac{2+\rho^k}{2-\rho^k}, C_2].$$

*(iii) If $\frac{\eta}{L^2}\|\nabla f(x_0)\| < \min(\frac{\rho^k(1-\rho^k)(2-\rho^k)}{3k^2(2+\rho^k)^2}, \frac{\rho^k-\rho_*}{3k^2C_*^2})$ then*

$$\|\nabla f(x_e)\| < \rho^k \|\nabla f(x_0)\| \text{ for } C \in [\tfrac{2+\rho^k}{2-\rho^k}, C_*].$$

*Proof.* Using Proposition 2.3.2, the result follows from upper bounding

$$\hat{\rho}(C) := \tilde{\rho}(C) + 3\tfrac{\eta}{L^2}k^2C^2\|\nabla f(x_0)\|,$$

by $\rho^k$. In addition, since $\tilde{\rho}$ is convex in $C$ (see Proposition 2.4.3) so is $\hat{\rho}$.

The case (i) follows directly from the fact that $\tilde{\rho}(C_1) = \rho_1 = \frac{\rho^k}{2-\rho^k}$ (see Lemma 2.4.6). For (ii) (resp. (iii)), we have $\tilde{\rho}(C_2) \leq \tilde{\rho}_{bo}(C_2) = \rho_2$ (resp. $\tilde{\rho}(C_*) = \rho_*$), thus taking $\frac{\eta}{L^2}\|\nabla f(x_0)\| < \min(\frac{\rho^k(1-\rho^k)(2-\rho^k)}{3k^2(2+\rho^k)^2}, \frac{\rho^k-\rho_2}{3k^2C_2^2})$ (resp. $\frac{\eta}{L^2}\|\nabla f(x_0)\| < \min(\frac{\rho^k(1-\rho^k)(2-\rho^k)}{3k^2(2+\rho^k)^2}, \frac{\rho^k-\rho_*}{3k^2C_*^2})$) implies $\hat{\rho}(C_1) < \rho^k$ and $\hat{\rho}(C_2) < \rho^k$ (resp. $\hat{\rho}(C_*) < \rho^k$), which gives the result using convexity of $\hat{\rho}$. ∎

Figure 2.4 displays the values of bounds from (2.16) with fixed $k, \mu$ and $L$ for various values of the perturbation parameter $\eta\|\nabla f(x_0)\|$. We observe that we do not loose much by using the simple upper bound with $M = 2$ compared with the numerical value of $\tilde{\rho}(C)$ obtained by solving (2.10). In the next section we study a version of Algorithm 2.2 with restarts.

### 2.5.2 Guarded and adaptive methods

Due to the particular form of the perturbation parameter $\alpha$, proportional to $\eta\|\nabla f(x_0)\|$ in the case of the gradient step operator, we see that as soon as $\eta\|\nabla f(x_0)\|$ is small enough to get $\tilde{\rho}(C) + 3\frac{\eta}{L^2}k^2C^2\|\nabla f(x_0)\| < 1$, one can restart Algorithm 2.2 to get a decreasing sequence of perturbation parameters, leading to faster convergence guarantees. Adding a *guarded step* to this scheme produces Algorithm 2.5.2. The guarded step consists in using the extrapolated point $x_e$ only if the gradient norm at this point is smaller than those of previous iterates, yielding global convergence guarantees.

**Guarded Constrained Anderson acceleration** (Algorithm 2.5.2)

**Input:**

- Objective function: $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ with $0 < \mu \leq L$.
- Initial guess: $x_0 \in \mathbb{R}^d$.
- Number of iterates used in extrapolation: $k \in \mathbb{N}^*$.
- Number of iterations: $N \in \mathbb{N}^*$.

**Run:**

For $i = 0, \ldots, N-1$:

$$x_i^0 = x_i$$

For $j = 0, \ldots, k$:

$$x_i^{j+1} = x_i^j - \tfrac{1}{L}\nabla f(x_i^j)$$

End For

Form $R = \begin{bmatrix} x_i^0 - x_i^1 & \cdots & x_i^k - x_i^{k+1} \end{bmatrix}$

Compute $\tilde{c} = \underset{\mathbf{1}^T c = 1,\ \|c\|_1 \leq C_{(i)}}{\operatorname{argmin}} \|Rc\|$

$$x_i^e = \sum_{j=0}^{k} \tilde{c}_j x_i^j$$

$$x_{i+1} = \underset{x \in \{x_i^e, x_i^k\}}{\operatorname{argmin}} \|\nabla f(x)\| \quad \textit{(guarded step)}$$

End For

**Output:** $x_N$

**Proposition 2.5.2.** *Let $f$ be a $\mu$-strongly convex function, with $L$-Lipschitz gradient and $\eta$-Lispchitz Hessian. Let $(x_i)_i \in \mathbb{R}^d$ be the sequence of iterates of Algorithm 2.5.2 on $f$, initiated at $x_0 \in \mathbb{R}^d$ with $k > 2$, $N \geq 1$ and with parameters $(C_{(i)})_i$ such that $C_{(i)} \geq 1$. It holds that*

$$\|\nabla f(x_N)\| \leq \prod_{i=1}^{N} \hat{\rho}_i(C_{(i-1)}) \|\nabla f(x_0)\|,$$

*where*

$$\hat{\rho}_i(C) = \min\left(\rho^k, \tilde{\rho}_{bo}(C) + 3\tfrac{\eta}{L^2}\|\nabla f(x_{i-1})\| k^2 C^2\right), \tag{2.17}$$

*with $\tilde{\rho}_{bo}$ defined in (2.14) with $\rho = 1 - \frac{\mu}{L}$.*

*Proof.* This is a direct consequence of Proposition 2.5.1. ∎

Using the explicit expression (2.16), one can for instance get a (conservative) lower bound on the number of iterations of Algorithm 2.5.2 for acceleration to occur (i.e. escape the guarded regime, which does nothing more than just iterating $F$).

**Corollary 2.5.3.** *Let $f$ be a $\mu$-strongly convex function, with $L$-Lipschitz gradient and $\eta$-Lispchitz Hessian. Let $(x_i)_i \in \mathbb{R}^d$ be the sequence of iterates of Algorithm 2.5.2 on $f$, initiated at $x_0 \in \mathbb{R}^d$ with $k > 2$, $N \geq 1$ and with parameters $(C_{(i)})_i$ such that $C_{(i)} \in [3, C_*]$. It holds that*

$$N \geq \frac{\log\left(\frac{\eta}{L^2}\frac{3k^2(2+\rho^k)^2\|\nabla f(x_0)\|}{\rho^k(1-\rho^k)(2-\rho^k)}\right)}{k\log\frac{1}{\rho}} \implies \prod_{i=1}^{N}\hat{\rho}_i(C_{(i-1)}) < \rho^{kN},$$

*with $\hat{\rho}_i(C_{(i-1)})$ defined in (2.17).*

*Proof.* We use (iii) from Proposition 2.5.1 along with $\frac{2+\rho^k}{2-\rho^k} \leq 3$. ∎

We notice that choosing $(C_{(i)})_i$ such that $\|\nabla f(x_{i-1})\|C_{(i-1)}^2$ tends to 0 with $i$ makes the perturbation terms $3\frac{\eta}{L^2}\|\nabla f(x_{i-1})\|k^2C_{(i-1)}^2$ in the convergence rate of Proposition 2.5.1 vanish with iterations. In addition, when the sequence $(C_{(i)})_i$ is unbounded above, there exists a rank $i_0$ such that $\tilde{\rho}_{bo}(C_{(i)}) = \rho_* \ \forall i \geq i_0$. Satisfying these two properties simultaneously would guarantee that $\hat{\rho}_i\left(C_{(i-1)}\right) \underset{i\to+\infty}{\longrightarrow} \rho_*$ (with $\hat{\rho}_i$ defined in (2.17)). We propose such an adaptive choice of $(C_{(i)})_i$ in the next corollary.

**Corollary 2.5.4.** *Under the conditions and notations of Proposition 2.5.2, with $(C_{(i)})_i$ satisfying*

$$C_{(i)} = i\left(\frac{L}{\|\nabla f(x_i)\|}\right)^{\delta} \quad \text{for } i \in \mathbb{N}, \tag{Adapt-Ctr}$$

*with $0 < \delta < \frac{1}{2}$, we have that*

$$\hat{\rho}_N(C_{(N-1)}) \underset{N\to+\infty}{\longrightarrow} \rho_*,$$

*meaning that asymptotically we reach the convergence rate of unconstrained Anderson acceleration on quadratics.*

*Proof.* With this choice of $C_{(i)}$ we have

$$\hat{\rho}_i(C_{(i-1)}) = \min\left(\rho^k, \tilde{\rho}_{bo}(C_{(i-1)}) + 3\frac{\eta}{L^{2(1-\delta)}}\|\nabla f(x_{i-1})\|^{1-2\delta}k^2\right).$$

We have $\|\nabla f(x_{i-1})\|$ that goes to 0 when $i$ grows, which implies that $C_{(i-1)}$ tends to $+\infty$ and thus $\tilde{\rho}_{bo}((C_{(i-1)})) = \rho_*$ for $i$ large enough. The choice $0 < \delta < \frac{1}{2}$ finally leads to the desired conclusion. ∎

The next section is dedicated to numerical testing of Algorithm 2.5.2 with the choice of constraints parameters (Adapt-Ctr).

### 2.5.3 Numerical experiments

For solving (2.5), we consider the following reformulation

$$\min_{\substack{\mathbf{1}^T c=1 \\ \|c\|_1 \leq C}} \tfrac{1}{2}\|Rc\|^2, \tag{2.18}$$

which we solve using a Frank-Wolfe method [Frank and Wolfe, 1956, Jaggi, 2013]. Indeed, as the constraint set is the convex hull of the set of points $\{\frac{C+1}{2}e_i + \frac{1-C}{2}e_j, i, j = 1, \ldots, k+1,\ i \neq j\}$ where $e_i$ is the unit vector of $\mathbb{R}^{k+1}$ with a one at the $i$-th position and zeros elsewhere. Frank-Wolfe methods have the advantage to offer simple access to an upper bound of the primal gap which is the stopping criterion we are interested in (see Corollary 2.3.3).

Figure 2.5 contains experiments performed on $\ell_2$ regularized logisitc regression. Blue curves correspond to gradient descent with step size $\frac{1}{L}$ where $L$ is the Lipschitz constant of the objective function. Red curves are obtained with Algorithm 2.5.2 using $C_i = +\infty$ (i.e. Anderson acceleration) and in that case (2.5) only involves solving a linear system. Finally, green curves correspond to Algorithm 2.5.2 using constraint parameters (Adapt-Ctr) with $\delta = 0.49$ (CAA).

Using an unconstrained or unregularized version of Anderson acceleration is often the best practical choice, although it is not generically guaranteed to even converge in all situations beyond quadratic minimization. We observe on Figure 2.5 that our constrained version (CAA) which provably guarantees acceleration exhibits similar good practical performances.

**Code** The implementation of CAA that we used for numerical experiments of Section 2.5.3 is available at

`https://github.com/mathbarre/ConstrainedAndersonAcceleration`

## Conclusion

In this chapter, we proposed upper bounds on the optimal value of a constrained Chebyshev problem, and used them to produce explicit, dimension independent, local convergence bounds on constrained Anderson acceleration applied to nonlinear operators with a particular emphasis on gradient step operators. In this setting, we proposed a guarded method with an adaptive choice of constraint parameter. Our convergence bounds are somewhat conservative as they rely on treating the nonlinear part of the operator as a perturbation of the linear setting. Some open questions remain. Can we remove the symmetry requirements in Assumptions 2.2.1 and 2.2.2 and still use a constrained Chebyshev arguments? Can we prove better convergence bounds on Anderson acceleration without decoupling linear and nonlinear parts of the operator? This last part would however require very different proof techniques.

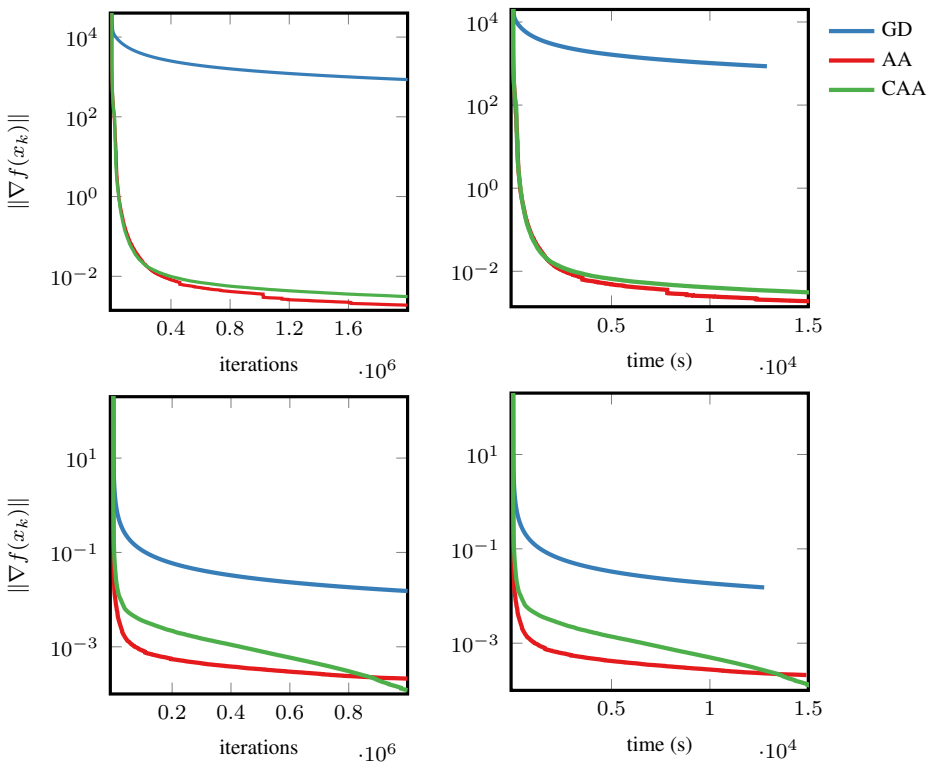


Figure 2.5: Comparison of Gradient descent (GD), vanilla Anderson acceleration (AA) and constrained Anderson acceleration with adaptive constraints parameters (CAA) on Logistic regression with $\ell_2$ regularization fixed to $10^{-8}L$ where $L$ is the Lipschitz constant of the logisitc regression. Top: Madelon dataset. Bottom: RCV1 dataset. Datasets are taken from the LIBSVM library [Chang and Lin, 2011].

# Chapter 3

# Problem Classes, Interpolation Theorems and Performance Estimation Problems

The purpose of this chapter is to introduce problems and notions that will be frequently mentioned and used in the last three chapters of this manuscript. We start by a short description of several standard functional classes used in the optimization literature. These classes are typically describe by sets of inequalities and we are particularly interested in their interpolation properties. These properties aim at characterizing functions in a particular class that interpolate a finite number of points using a finite number of inequalities. Then, we describe the Performance Estimation Problem (PEP) framework initiated by Drori and Teboulle [2014] and further developed by Taylor et al. [2017c] for analyzing optimization methods in a principled way.

**Codes** Matlab scripts and Mathematica notebooks can be found at

`https://github.com/mathbarre/Chapter3_manuscript`,

to help the reader reproducing the examples and proofs of Section 3.2.

## 3.1 Functional classes and interpolation theorems

In this section, we review some classical class of problems on which first-order optimization algorithm are typically applied. Together with their definitions we detail some interpolation (or extension) results on these classes that will be used many times in this manuscript. We refer to classical textbooks on convex analysis as Rockafellar [1996], Hiriart-Urruty and Lemaréchal [2013] for more details on the functional classes and to Taylor [2017, Section 3] for more developments on interpolation.

### 3.1.1 Closed convex proper functions

We start be the definition of closed, convex and proper functions which constitute a class of particular interest in optimization.

**Definition 3.1.1** (Closed convex proper functions)**.** *Let $d \in \mathbb{N}^*$, we denote by $\mathcal{F}_{0,\infty}(\mathbb{R}^d)$ the set of closed convex proper functions on $\mathbb{R}^d$ defined as*

$$\mathcal{F}_{0,\infty}(\mathbb{R}^d) = \left\{ \begin{array}{ll} f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\} \text{ such that} & \\ \forall \alpha \in \mathbb{R},\ \{x \in \mathbf{dom}\, f,\ f(x) \leq \alpha\} \text{ is a closed set,} & (closed) \\ \forall x, y, \mathbb{R}^d,\ \forall t \in [0,1],\ f((1-t)x + ty) \leq (1-t)f(x) + tf(y), & (convex) \\ \mathbf{dom}\, f \neq \emptyset & (proper) \end{array} \right\},$$

*where* $\mathbf{dom}\, f = \{x \in \mathbb{R}^d, f(x) < +\infty\}$.

When minimizing such functions, one can characterize their optimal points using a "first-order" optimality condition relying the notions of subdifferentials.

**Definition 3.1.2** (Subdifferentials)**.** *Let $d \in \mathbb{N}^*$, $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ and $x \in \mathbf{dom}\, f$, the subdifferential of $f$ at $x$ is defined as*

$$\partial f(x) = \{g \in \mathbb{R}^d, \forall y \in \mathbb{R}^d\, f(y) \geq f(x) + \langle g, x - y \rangle\}.$$

When they exist, subdifferentials define affine lower bounds on the considered function. The following theorem guarantees that closed convex proper functions admit subdifferential at each point in their domain.

**Theorem 3.1.3.** *Let $d \in \mathbb{N}^*$ and $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$, the following equivalence holds*

$$f \text{ is proper and convex } \implies \forall x \in \mathrm{ri}(\mathbf{dom}\, f),\ \partial f(x) \neq \emptyset.$$

*Proof.* See e.g. Rockafellar [1996, Theorem 23.4]. ■

The previous theorem connects convexity of $f$ with the existence of affine lower bounds at all point of $\mathbf{dom}\, f$, and in particular we can derive the following optimality condition for a point $x_* \in \mathbf{dom}\, f$ to be a minimum of $f$.

**Corollary 3.1.4.** *Let $d \in \mathbb{N}^*$ and $f \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$, the following equivalence holds*

$$x_* \in \operatorname*{argmin}_{x \in \mathbb{R}^d} f(x) \iff 0 \in \partial f(x_*).$$

*Proof.* Both sides can be reformulated as $f(x_*) \leq f(x)$, $\forall x \in \mathbb{R}^d$. ■

Together with this class of function, we present some convex interpolation results that sre crucial tools in the study of worst-case analysis methodology detailed in the second part of this chapter. The interpolation problem can be stated as follow: given a finite index set $I$ and a finite set of triplets $S = \{(x_i, g_i, f_i)\}_{i \in I} \subset \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}$, does there exist a function $f \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ such that $g_i \in \partial f(x_i)$ and $f_i = f(x_i)$ for all $i \in I$ ? The following theorem provide necessary and sufficient condition for such a function to exist. We give a proof of this theorem as it show how to construct such an interpolating function.

**Theorem 3.1.5** (Theorem 1 of Taylor et al. [2017c])**.** *Let $d \in \mathbb{N}^*$, $I$ be a finite index set and $S = \{(x_i, g_i, f_i)\}_{i \in I} \subset \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}$. The following equivalence holds*

$$\exists f \in \mathcal{F}_{0,\infty}(\mathbb{R}^d) \text{ such that } g_i \in \partial f(x_i) \text{ and } f_i = f(x_i) \text{ for all } i \in I, \tag{3.1}$$

*if and only if*

$$f_j \geq f_i + \langle g_i, x_j - x_i \rangle, \text{ for all } (x_i, g_i, f_i), (x_j, g_j, f_j) \in S. \tag{3.2}$$

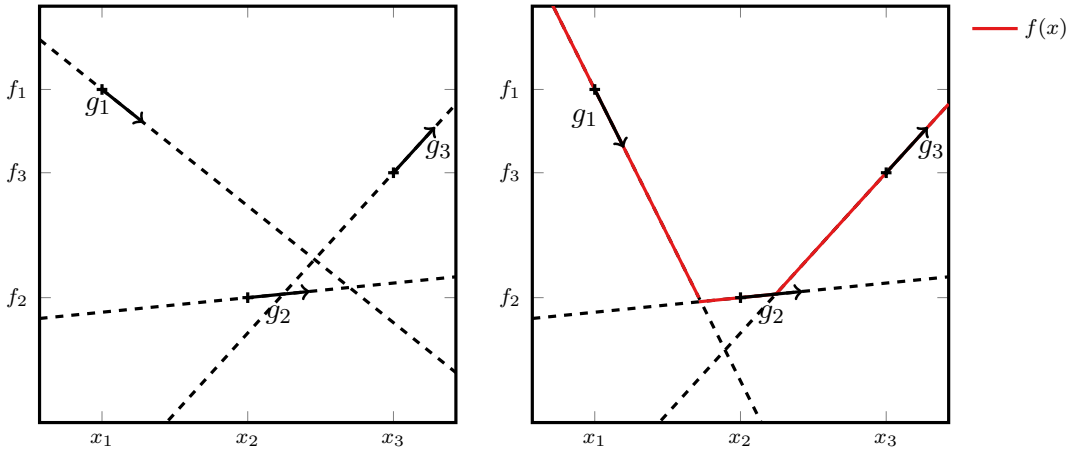


Figure 3.1: On the left hand side, interpolation conditions (3.2) are not satisfied as $f_2 < f_1 + \langle g_1, x_2 - x_1 \rangle$, therefore we can note construct a convex interpolation function as it should stay above all the dashed curves and still be equal to $f_2$ in $x_2$. On the right hand side, conditions (3.2) are satisfied and we can construct a convex interpolation function $f$.

*Proof.* ( $\Longrightarrow$ ) follows from Definition 3.1.2 on subdifferentials.
For ( $\Longleftarrow$ ), we observe that the function

$$f(x) = \max_{(x_i, g_i, f_i) \in S} f_i + \langle g_i, x - x_i \rangle,$$

is in $\mathcal{F}_{0,\infty}(\mathbb{R}^d)$, $f(x_i) = f_i$ and $g_i \in \partial f(x_i)$. ■

We illustrate this result on some 1-dimensional examples in Figure 3.1.

**Strongly convex functions** A standard restriction of the closed convex proper function class is constituted by convex functions that stay convex when one subtracts a convex quadratic term from it. More precisely, the class of $\mu$-strongly convex functions is defined as follows.

**Definition 3.1.6** (Strong convexity)**.** *Let $d \in \mathbb{N}^*$ and $\mu \geq 0$, we denote by $\mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$ the set of closed convex proper functions on $\mathbb{R}^d$ defined as*

$$\mathcal{F}_{\mu,\infty}(\mathbb{R}^d) = \left\{ f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\} \text{ s.t. } f(\cdot) - \tfrac{\mu}{2}\|\cdot\|^2 \in \mathcal{F}_{0,\infty}(\mathbb{R}^d) \right\}.$$

From this definition, one can directly extend the results of Theorem 3.1.5 to provide interpolation result on strongly convex functions.

**Theorem 3.1.7** (Strongly convex interpolation)**.** *Let $d \in \mathbb{N}^*$, $\mu \geq 0$, $I$ be a finite index set and $S = \{(x_i, g_i, f_i)\}_{i \in I} \subset \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}$. The following equivalence holds*

$$\exists f \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d) \text{ such that } g_i \in \partial f(x_i) \text{ and } f_i = f(x_i) \text{ for all } i \in I, \tag{3.3}$$

*if and only if*

$$f_j \geq f_i + \langle g_i, x_j - x_i \rangle + \tfrac{\mu}{2}\|x_i - x_j\|^2, \text{ for all } (x_i, g_i, f_i), (x_j, g_j, f_j) \in S. \tag{3.4}$$

*Proof.* The result follows from Theorem 3.1.5 applied to $S_\mu = \{(x_i, g_i - \mu x_i, f_i - \frac{\mu}{2}\|x_i\|^2)\}_{i \in I}$ which guarantee the existence of $\tilde{f} \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ such that $g_i - \mu x_i \in \partial \tilde{f}(x_i)$ and $f_i - \frac{\mu}{2}\|x_i\|^2 = \tilde{f}(x_i)$, and therefore $f(\cdot) = \tilde{f}(\cdot) + \frac{\mu}{2}\|\cdot\|^2 \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$ suits. ■

### 3.1.2 Smooth and convex functions

When the closed convex and proper objective function to be minimized is differentiable, the subdifferential sets are singletons (i.e., $\partial f(x) = \{\nabla f(x)\}$) and a standard assumptions to ensure convergence of many first-order methods (e.g., gradient descent with fixed step size) is to require the gradient to be Lipschitz (Lipschitzness of the gradient is also referred as smoothness). We define the class of closed convex and proper functions with Lipschitz subgradient mapping as follows.

**Definition 3.1.8** (Smoothness and convexity)**.** *Let $d \in \mathbb{N}^*$ and $L \geq 0$, we denote by $\mathcal{F}_{0,\mathrm{L}}(\mathbb{R}^d)$ the set of functions on $\mathbb{R}^d$ defined as*

$$\mathcal{F}_{0,\mathrm{L}}(\mathbb{R}^d) = \left\{ \begin{array}{l} f \in \mathcal{F}_{0,\infty}(\mathbb{R}^d) \text{ s.t.} \forall x, y \in \mathbf{dom}\, f \text{ and } \forall s_f(x), s_f(y) \in \partial f(x), \partial f(y), \\ \frac{1}{L}\|s_f(x) - s_f(y)\| \leq \|x - y\| \end{array} \right\}.$$

Note that this definition does not directly use the gradient of $f$ and is therefore valid when $L = \infty$. In the following lemmas, we see that for $L < +\infty$ we recover the class of convex function with $L$-Lipschtiz gradient. First let us prove that elements $f \in \mathcal{F}_{0,\mathrm{L}}(\mathbb{R}^d)$ necessarily satisfy $\mathbf{dom}\, f = \mathbb{R}^d$ when $L$ is finite.

**Lemma 3.1.9.** *Let $d \in \mathbb{N}^*$, $0 \leq L < +\infty$ and $f \in \mathcal{F}_{0,\mathrm{L}}(\mathbb{R}^d)$, it holds that*

$$\mathbf{dom}\, f = \mathbb{R}^d.$$

*Proof.* First, let us notice several useful facts. It follows from Definition 3.1.8 and Theorem 3.1.3 that elements of $\mathrm{ri}(\mathbf{dom}\, f)$ have unique subgradients and that the application $x \to s_f(x)$ is $L$-Lipschitz, and therefore continuous.

In addition, let $x, y \in \mathrm{ri}(\mathbf{dom}\, f)$, Definition 3.1.2 allows writing for $s_f(y) \in \partial f(y)$

$$f(y) + \langle s_f(y), x - y\rangle \leq f(x),$$

which implies that

$$f(y) + \langle s_f(y) - s_f(x), x - y\rangle \leq f(x) + \langle s_f(x), y - x\rangle,$$

with $s_f(x) \in \partial f(x)$. Using the smoothness Definition 3.1.8 and Cauchy-Schwarz inequality we obtain

$$f(y) \leq f(x) + \langle s_f(x), y - x\rangle + L\|x - y\|^2. \tag{3.5}$$

Now, assume that $\mathbf{dom}\, f \neq \mathbb{R}^d$. This implies that $\mathrm{ri}(\mathbf{dom}\, f) \neq \mathbb{R}^d$ and in particular $\partial(\mathrm{ri}(\mathbf{dom}\, f))$ is not empty. We denote by $x_\partial$ an elements of this boundary.

There exists a sequence $(x_k)_k$ of elements of $\mathrm{ri}(\mathbf{dom}\, f)$ such that $x_k \to x_\partial$. Note that (3.5) implies that $(f(x_k))_k$ is bounded (as $x_k$ is a converging sequence) and Definition 3.1.8 implies that $(s_f(x_k))_k$ is also bounded. Therefore, one can find an extraction $\phi$, a vector $g_\partial \in \mathbb{R}^d$ and a real $f_\partial \in \mathbb{R}$ such that $f(x_{\phi(k)}) \to f_\partial$ and $s_f(x_{\phi(k)}) \to g_\partial$. By convexity, Theorem 3.1.3 allows writing for all $x \in \mathbf{dom}\, f$,

$$f(x_{\phi(k)}) + \langle s_f(x_{\phi(k)}), x - x_{\phi(k)}\rangle \leq f(x),$$

and by taking the limit in $k$, we get

$$f_\partial + \langle g_\partial, x - x_\partial\rangle \leq f(x),$$

and using the lower semicontinuity of closed convex proper function [Rockafellar, 1996, Theorem 7.1] gives $f(x_\partial) \leq f_\partial$ which leads to

$$f(x_\partial) + \langle g_\partial, x - x_\partial \rangle \leq f(x),$$

for all $x \in \mathbf{dom}\, f$. Finally, $g_\partial \in \partial f(x_\partial)$ and using the supporting hyperplane theorem on $\mathrm{ri}(\mathbf{dom}\, f)$ at $x_\partial$ provides a nonzero direction $v \in \mathbb{R}^d$ such that

$$\langle v, x - x_\partial \rangle \leq 0 \quad \text{for all } x \in \mathrm{ri}(\mathbf{dom}\, f),$$

and this is also true for all $x \in \mathbf{dom}\, f$ (as the adherence of $\mathrm{ri}(\mathbf{dom}\, f)$ contains $\mathbf{dom}\, f$). This implies that $g_\partial + \lambda v$ are subgradients for all $\lambda \geq 0$, and is in contradiction with the unicity of the subgradients induced by Definition 3.1.8. Therefore, $\partial(\mathrm{ri}(\mathbf{dom}\, f))$ has to be empty, and $\mathrm{ri}(\mathbf{dom}\, f) = \mathbb{R}^d$.

■

Differentiability of the function $f \in \mathcal{F}_{0,\mathrm{L}}(\mathbb{R}^d)$ when $L < +\infty$ is ensured by the following lemma.

**Lemma 3.1.10.** *Let $d \in \mathbb{N}^*$, $0 \leq L < +\infty$ and $f \in \mathcal{F}_{0,\mathrm{L}}(\mathbb{R}^d)$, then $f$ is continuously differentiable and its gradient satisfies*

$$\forall x, y \in \mathbb{R}^d,\ \|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|.$$

*Proof.* First note that using Lemma 3.1.9, it holds that $\mathbf{dom}\, f = \mathbb{R}^d$. We start by showing that $\partial f(x)$ is reduced to a singleton for all $x \in \mathbb{R}^d$. Indeed, choosing $x = y$ in Definition 3.1.8 provides us the desired conclusion, and we use the notation $s_f(x)$ to refer to the unique element of $\partial f(x)$.

Then, we show that $f$ is differentiable and that $s_f(x)$ corresponds to the gradient of $f$ at $x$. For $v \in \mathbb{R}^d$ with $\|v\| = 1$ and $t > 0$, Definition 3.1.2 allows obtaining the following inequality

$$f(x) + t\langle s_f(x), v \rangle \leq f(x + tv) \leq f(x) + t\langle s_f(x + tv), v \rangle. \tag{3.6}$$

From Lipschitzness in Definition 3.1.8, we have that

$$\|s_f(x) - s_f(x + tv)\| \leq tL,$$

which shows that $s_f(x + tv) \underset{t \to 0}{\to} s_f(x)$, and implies together with (3.6) that

$$\frac{f(x + tv) - f(x)}{t} \underset{t \to 0}{\to} \langle s_f(x), v \rangle.$$

Therefore $f$ is differentiable and $\nabla f(x) = s_f(x)$ which concludes the proof. ■

The next theorem lists some standard characterizations of convex function with Lipschitz gradient (see e.g. Nesterov [2018, Theorem 2.1.5]) that will be used in the rest of the manuscript.

**Theorem 3.1.11.** *Let $d \in \mathbb{N}^*$, $f \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ and differentiable, the following properties are equivalent*

*(i)* $f \in \mathcal{F}_{0,\mathrm{L}}(\mathbb{R}^d)$,

*(ii)* $\forall x, y \in \mathbb{R}^d,\ \|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$,

*(iii)* $\forall x, y \in \mathbb{R}^d,\ f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|x - y\|^2$,

*(iv)* $\forall x, y \in \mathbb{R}^d$, $\langle \nabla f(x) - \nabla f(y), x - y \rangle \leq L\|x - y\|^2$,

*(v)* $\forall x, y \in \mathbb{R}^d$, $f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|^2 \leq f(y)$,

*(vi)* $\forall x, y \in \mathbb{R}^d$, $\frac{1}{L}\|\nabla f(x) - \nabla f(y)\|^2 \leq \langle \nabla f(x) - \nabla f(y), x - y \rangle$.

In order to provide simple proofs of interpolation theorems for smooth convex functions, let us introduce the notion of Fenchel conjugate of a convex function [Rockafellar, 1996, §12].

**Definition 3.1.12** (Convex Conjugate)**.** *Let $d \in \mathbb{N}^*$ and $f : \mathbb{R}^d \to \mathbb{R} \cup \{-\infty, +\infty\}$. The Fenchel conjugate of $f$ is denoted $f^*$ and is defined as*

$$\forall x \in \mathbb{R}^d,\ f^*(x) = \sup_{y \in \mathbb{R}^d} \langle x, y \rangle - f(y).$$

In the following theorem we list important properties of the Fenchel conjugate of closed convex and proper functions [Rockafellar, 1996, §12, §26] that we use in the sequel.

**Theorem 3.1.13.** *Let $d \in \mathbb{N}^*$ and $f \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$, the following properties hold*

*(i)* $f^* \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$,

*(ii)* $(f^*)^* = f$,

*(iii)* *Let $x \in \mathbf{dom}\, f$:* $g \in \partial f(x) \iff x \in \partial f^*(g)$,

*(iv)* *Let $x \in \mathbf{dom}\, f$ and $g \in \partial f(x)$, then* $f^*(g) = \langle g, x \rangle - f(x)$,

*(v)* *Let $L \geq 0$,* $f \in \mathcal{F}_{0,\mathrm{L}}(\mathbb{R}^d) \iff f^* \in \mathcal{F}_{1/L,\infty}(\mathbb{R}^d)$.

Theorem 3.1.13 (v) links the smoothness property to strong convexity using Fenchel conjugate. This relation allows converting the interpolation results of Theorem 3.1.7 into interpolation results for closed convex and proper functions with Lipschitz gradients.

**Theorem 3.1.14** (Smooth and convex interpolation)**.** *Let $d \in \mathbb{N}^*$, $L \geq 0$, $I$ be a finite index set and $S = \{(x_i, g_i, f_i)\}_{i \in I} \subset \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}$. The following equivalence holds*

$$\exists f \in \mathcal{F}_{0,\mathrm{L}}(\mathbb{R}^d) \text{ such that } g_i \in \partial f(x_i) \text{ and } f_i = f(x_i) \text{ for all } i \in I, \tag{3.7}$$

*if and only if*

$$f_j \geq f_i + \langle g_i, x_j - x_i \rangle + \tfrac{1}{2L}\|g_i - g_j\|^2, \text{ for all } (x_i, g_i, f_i), (x_j, g_j, f_j) \in S. \tag{3.8}$$

*Proof.* Applying Theorem 3.1.7 with $\mu = \frac{1}{L}$ to the set $S_L = \{(g_i, x_i, \langle g_i, x_i \rangle - f_i)\}_{i \in I}$ leads to the following equivalence

$$\exists h \in \mathcal{F}_{1/L,\infty}(\mathbb{R}^d) \text{ such that } x_i \in \partial h(g_i) \text{ and } h(g_i) = \langle g_i, x_i \rangle - f_i \text{ for all } i \in I, \tag{3.9}$$

if and only if

$$f_i \geq f_j + \langle g_j, x_i - x_j \rangle + \tfrac{1}{2L}\|g_j - g_i\|^2, \text{ for all } (x_i, g_i, f_i), (x_j, g_j, f_j) \in S.$$

Finally, Theorem 3.1.13 allows writing

$$(3.9) \iff \exists f \in \mathcal{F}_{0,\mathrm{L}}(\mathbb{R}^d) \text{ such that } g_i \in \partial f(x_i) \text{ and } f(x_i) = f_i \text{ for all } i \in I,$$

using convex conjugate, and this concludes the proof. ■

We end this first section by studying the class of functions that are both smooth and strongly convex.

### 3.1.3 Smoothness and strong convexity

Another, class of functions that is often considered in convex optimization literature is the class of closed convex proper functions that are strongly convex with Lipschitz gradients. This class contains for instance quadratic functions with positive definite Hessians.

**Definition 3.1.15** (Smoothness and strongly convexity)**.** *Let $d \in \mathbb{N}^*$ and $0 \leq \mu \leq L$, we denote by $\mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$ the set of functions on $\mathbb{R}^d$ defined as*

$$\mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d) = \mathcal{F}_{\mu,\infty}(\mathbb{R}^d) \cap \mathcal{F}_{0,\mathrm{L}}(\mathbb{R}^d).$$

Note that for $\mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$ to be nonempty, $\mu$ has to be smaller than $L$, indeed from Theorem 3.1.3 applied to $f(\cdot) - \frac{\mu}{2}\|\cdot\|^2$ and from Definition 3.1.8, one can show that

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

In order to get interpolation conditions for $\mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$, we first need the following lemma.

**Lemma 3.1.16** (Theorem 3 of Taylor et al. [2017c])**.** *Let $d \in \mathbb{N}^*$ and $0 \leq \mu \leq L$, the following equivalence holds*

$$f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d) \iff f(\cdot) - \tfrac{\mu}{2}\|\cdot\|^2 \in \mathcal{F}_{0,L-\mu}(\mathbb{R}^d).$$

*Proof.* The case $L = +\infty$ is trivial, therefore we consider $L < +\infty$. From Theorem 3.1.11 we have that $f(\cdot) - \frac{\mu}{2}\|\cdot\|^2 \in \mathcal{F}_{0,L-\mu}(\mathbb{R}^d)$ is equivalent to

$$\forall x, y \in \mathbb{R}^d,\ \langle \nabla f(x) - \nabla f(y) - \mu(x - y), x - y\rangle \leq (L - \mu)\|x - y\|^2,$$

which can be reformulated as

$$\forall x, y \in \mathbb{R}^d,\ \langle \nabla f(x) - \nabla f(y), x - y\rangle \leq L\|x - y\|^2,$$

and which is equivalent to $f \in \mathcal{F}_{0,\mathrm{L}}(\mathbb{R}^d)$ using once again Theorem 3.1.11.

In addition, $f(\cdot) - \frac{\mu}{2}\|\cdot\|^2 \in \mathcal{F}_{0,L-\mu}(\mathbb{R}^d) \subset \mathcal{F}_{0,\infty}(\mathbb{R}^d)$, therefore $f \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$ by Definition 3.1.6 and we get the desired conclusion.

■

Lemma 3.1.16 allows using directly Theorem 3.1.14 and obtaining the following interpolation conditions.

**Theorem 3.1.17** (Theorem 4 of Taylor et al. [2017c])**.** *Let $d \in \mathbb{N}^*$, $0 \leq \mu \leq L$, $I$ be a finite set of indexes and $S = \{(x_i, g_i, f_i)\}_{i\in I} \subset \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}$. The following equivalence holds*

$$\exists f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d) \text{ such that } g_i \in \partial f(x_i) \text{ and } f_i = f(x_i) \text{ for all } i \in I, \tag{3.10}$$

*if and only if*

$$f_j \geq f_i + \langle g_i, x_j - x_i\rangle + \tfrac{1}{2(L-\mu)}\|g_i - g_j - \mu(x_i - x_j)\|^2 + \tfrac{\mu}{2}\|x_i - x_j\|^2, \quad \text{for all } (x_i, g_i, f_i), (x_j, g_j, f_j) \in S. \tag{3.11}$$

*Proof.* We obtain the desired result by using Lemma 3.1.16 and applying Theorem 3.1.14 to $S_\mu = \{(x_i, g_i - \mu x_i, f_i - \frac{\mu}{2}\|x_i\|^2)\}_{i\in I}$ with smoothness parameter $L - \mu$. ■

**Remark 3.1.18.** *In the last three chapter, Theorem 3.1.7 is often used under the form*

$$f_j \geq f_i + \langle g_i, x_j - x_i\rangle + \tfrac{1}{2L}\|g_i - g_j\|^2 + \tfrac{\mu}{2\left(1-\frac{\mu}{L}\right)}\|x_i - x_j - \tfrac{1}{L}(g_i - g_j)\|^2 \quad \text{for all } (x_i, g_i, f_i), (x_j, g_j, f_j) \in S.$$

## 3.2 Introduction to performance estimation problems

In this section we present recent developments in worst-case analyses of optimization methods. The *Performance Estimation Problem* (PEP) methodology consists in expressing worst-case guarantees of an optimization algorithm as the result of another optimization problem which can be solved numerically using semidefinite programing. This idea, originate from Drori and Teboulle [2014] and has been further developed by Taylor et al. [2017c]. We refer to Section 1.1.3 of Chapter 1 for a short literature review. We start by presenting this framework in an abstract setting, and then develop it in the context of gradient methods applied to smooth convex functions.

Let $N \in \mathbb{N}$ and $R \in \mathbb{R}^*$, for all dimensions $d$, we define the following elements

- A sequence of functional spaces $\{\mathcal{F}_d\}_{d\in\mathbb{N}^*}$ with $\mathcal{F}_d \subset \mathcal{F}(\mathbb{R}^d, \mathbb{R}\cup\{-\infty,+\infty\})$,
- a sequence of optimization methods $\{\mathcal{M}_d\}_{d_\in\mathbb{N}^*}$ with $\mathcal{M}_d : \mathcal{F}\times\mathbb{R}^d\times\mathbb{N}\to\mathbb{R}^d$ (i.e., given a function $f\in\mathcal{F}_d$, a starting point $x_0\in\mathbb{R}^d$ and a number of iteration $k\in\mathbb{N}$, $x_k = \mathcal{M}_d(f,x_0,k)$ is the $k$-th iterate of the method),
- an objective criterion $\Phi_{d,\text{obj}} : \mathcal{F}_d\times(\mathbb{R}^d)^{N+1}\to\mathbb{R}\cup\{-\infty,+\infty\}$,
- an initial condition $\Phi_{d,\text{init}} : \mathcal{F}_d\times(\mathbb{R}^d)^{N+1}\to\mathbb{R}\cup\{-\infty,+\infty\}$.

We are interested in finding a bound $C(N,R)$ on the value (possibly $+\infty$) of $\Phi_{d,\text{obj}}(f,\{x_k\}_{k=0\dots N})$ for all dimension $d\in\mathbb{N}^*$, all input value $x_0\in\mathbb{R}^d$ , all function $f\in\mathcal{F}_d$, such that $x_k = \mathcal{M}_d(f,x_0,k)$ for $k=1\dots,N$ and such that $\Phi_{d,\text{init}}(f,\{x_k\}_{k=0\dots N})\leq R^2$. This can be expressed as the optimization problem

$$\begin{aligned} C(N,R) = \sup_{\substack{d,f\\ x_0\in\mathbb{R}^d}} \ & \Phi_{d,\text{obj}}(f,\{x_k\}_{k=0\dots N}) \\ \text{s.t. } & f\in\mathcal{F}_d, \\ & R^2 \geq \Phi_{d,\text{init}}(f,\{x_k\}_{k=0\dots N}), \\ & x_k = \mathcal{M}_d(f,x_0,k) \quad \text{for } k=1\dots,N. \end{aligned} \tag{3.12}$$

Without additional assumptions, solving problem (3.12) appears out of reach. It involves variables belonging to potentially infinite dimensional functional spaces and has many dependence in the dimension $d$. In the next section we impose some constraints on $\Phi_{d,\text{obj}}$, $\Phi_{d,\text{init}}$, and instantiate $\mathcal{F}_d$ and $\mathcal{M}_d$ in order to reformulate (3.12) as a tractable semidefinite program.

### 3.2.1 Performance estimation for gradient methods

For illustration purposes, we work with functional spaces $\mathcal{F}_d = \mathcal{F}_{\mu,\text{L}}(\mathbb{R}^d)$ with $0\leq\mu<L<+\infty$ defined in Definition 3.1.15 from this point on. In particular, we focus on analyzing worst-case performance of fixed-step (i.e. iterates are obtained as predefined combinations of the gradients) gradient methods, and other types of methods are analyzed in Chapters 4 to 6. These fixed-step methods correspond to fix a sequence of parameters $\{\gamma_{i,j}\}_{i,j\in\mathbb{N}}$ such that

$$\forall d\in\mathbb{N}^*,\ f\in\mathcal{F}_{\mu,\text{L}}(\mathbb{R}^d),\ x_0\in\mathbf{dom}\, f,\ k\in\mathbb{N},\ \mathcal{M}_d(f,x_0,k) = x_0 - \sum_{i=0}^{k-1}\gamma_{k,i}\nabla f(x_i). \tag{3.13}$$

In addition, we consider first-order methods that are not degenerate (i.e. use the last gradient available at each iteration), that is

$$\gamma_{k,k-1}\neq 0, \quad \text{for } k\geq 1. \tag{3.14}$$

We also assume $\Phi_{d,\text{obj}}$ (objective criterion) and $\Phi_{d,\text{init}}$ (initial condition) to be Gram-representable (or SDP representable) in the sense of the following definition.

**Definition 3.2.1** (Gram-representability)**.** *Let $d \in \mathbb{N}^*$, $N \in \mathbb{N}$, we say that $\Phi_{d,\text{obj}}$ and $\Phi_{d,\text{init}}$ are Gram-representable, if for $f \in \mathcal{F}_{\mu,\text{L}}(\mathbb{R}^d)$ and a sequence $\{x_k\}_{k=0,\dots,N}$, it holds that*

*$\Phi_{d,\text{obj}}(f, \{x_k\}_{k=0,\dots,N})$ and $\Phi_{d,\text{init}}(f, \{x_k\}_{k=0\dots N})$ are linear combinations of $f(x_*), f(x_0), \dots, f(x_N)$ and of $\langle x, y\rangle$ for $x, y \in \{x_k\}_{k\in\{*,0,\dots,N\}} \cup \{\nabla f(x_k)\}_{k\in\{*,0,\dots,N\}}$, where $x_* \in \operatorname{argmin}_x f(x)$. In addition, the linear combinations are independent of the dimension $d$.*

With these choices of functional spaces, methods, objective criteria and initial conditions, the optimization problem (3.12) can be reformulated as

$$\begin{aligned} C(N,R) = \sup_{\substack{d,f \\ x_0\in\mathbb{R}^d}} & \Phi_{d,\text{obj}}(S) \\ \text{s.t. } & f \in \mathcal{F}_{\mu,\text{L}}(\mathbb{R}^d), \\ & S = \{(x_k, \nabla f(x_k), f(x_k))\}_{k\in\{*,0,\dots,N\}}, \\ & \nabla f(x_*) = 0, \\ & R^2 \geq \Phi_{d,\text{init}}(S), \\ & x_k = x_0 - \sum_{i=0}^{k-1} \gamma_{k,i} \nabla f(x_i) \quad \text{for } k = 1\dots, N, \end{aligned} \tag{3.15}$$

where we use the notations $\Phi_{d,\text{obj}}(S)$ and $\Phi_{d,\text{init}}(S)$, as Definition 3.2.1 implies that these functions only depend on elements of $S$. We can observe that, apart from the functional constraint $f \in \mathcal{F}_{\mu,\text{L}}(\mathbb{R}^d)$, objective and constraints only depend on elements of the finite set $S$. We can slightly reformulate (3.15) as

$$\begin{aligned} C(N,R) = \sup_{d,\, S} & \Phi_{d,\text{obj}}(S) \\ \text{s.t. } & S = \{(x_k, g_k, f_k)\}_{k\in\{*,0,\dots,N\}}, \\ & \exists f \in \mathcal{F}_{\mu,\text{L}}(\mathbb{R}^d), g_k = \nabla f(x_k) \text{ and } f_k = f(x_k) \text{ for } k \in \{*, 0, \dots, N\}, \\ & g_* = 0, \\ & R^2 \geq \Phi_{d,\text{init}}(S), \\ & x_k = x_0 - \sum_{i=0}^{k-1} \gamma_{k,i} g_i \quad \text{for } k = 1\dots, N, \end{aligned} \tag{3.16}$$

and observe that we can use interpolation results from Theorem 3.1.17 to replace the existence condition of $f$ by a finite set of inequalities on the elements of $S$.

This leads us to

$$
\begin{aligned}
C(N,R) = \sup_{d,\,S}\ & \Phi_{d,\mathrm{obj}}(S) \\
\text{s.t. } & S = \{(x_k, g_k, f_k)\}_{k\in\{*,0,\dots,N\}}, \\
& f_j \geq f_i + \langle g_i, x_j - x_i\rangle + \tfrac{1}{2(L-\mu)}\|g_i - g_j - \mu(x_i - x_j)\|^2 \\
& \qquad + \tfrac{\mu}{2}\|x_i - x_j\|^2 \text{ for } i,j \in \{*,0,\dots,N\}, \\
& g_* = 0, \\
& R^2 \geq \Phi_{d,\mathrm{init}}(S), \\
& x_k = x_0 - \sum_{i=0}^{k-1} \gamma_{k,i} g_i \quad \text{for } k = 1\dots,N,
\end{aligned}
\tag{3.17}
$$

which has finite dimension. The maximization over the dimension $d$ remains a problem as it seems difficult to perform, however we can use the Gram-representability assumption (see Definition 3.2.1) to write (3.17) as a linear semidefinite program as detailed in the next section.

**Semidefinite reformulation** Given $S = \{(x_k, g_k, f_k)\}_{k\in\{*,0,\dots,N\}} \subset \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}$, we define the vector of function values

$$F = [f_*, f_0, \dots, f_N] \in \mathbb{R}^{1\times N+2},$$

and the Gram-matrix

$$G = X^T X \in \mathbf{S}^+_{N+3}, \tag{3.18}$$

with

$$X = [x_*, x_0, g_0, \dots, g_N] \in \mathbb{R}^{d\times N+3}.$$

We set Definition 3.2.1 down in order to be able writing $\Phi_{d,\mathrm{obj}}$ and $\Phi_{d,\mathrm{init}}$ in terms of $F$ and $G$. In particular, there exist $A_{\mathrm{obj}}, A_{\mathrm{init}} \in \mathbb{R}^{N+3\times N+3}$ and $a_{\mathrm{obj}}, a_{\mathrm{init}} \in \mathbb{R}^{N+2}$ such that for all $d \in \mathbb{N}^*$

$$\Phi_{d,\mathrm{obj}}(S) = \mathrm{Tr}(A_{\mathrm{obj}} G) + F a_{\mathrm{obj}},$$

and

$$\Phi_{d,\mathrm{init}}(S) = \mathrm{Tr}(A_{\mathrm{init}} G) + F a_{\mathrm{init}}.$$

In order to express (3.17) in terms of variables $F$ and $G$, we introduce the convenient notations $\{\mathbf{x}_k\}_{k\in\{*,0,\dots,N\}}$, $\{\mathbf{g}_k\}_{k\in\{*,0,\dots,N\}}$ and $\{\mathbf{f}_k\}_{k\in\{*,0,\dots,N\}}$ allowing to select the right entries in $F$ and $G$ as

$$
\begin{aligned}
x_k = X\mathbf{x}_k, g_k = X\mathbf{g}_k \\
f_k = F\mathbf{f}_k, \text{ for } k \in \{*,0,\dots,N\}.
\end{aligned}
$$

To do so, we set

$$
\begin{aligned}
&\mathbf{g}_* = 0,\ \mathbf{f}_* = u_1,\ \mathbf{x}_* = e_1,\ \mathbf{x}_0 = e_2, \\
&\mathbf{f}_k = u_{k+2},\ \mathbf{g}_k = e_{k+3},\ \text{ for } k = 0,\dots,N,
\end{aligned}
$$

where $\{e_k\}_{k=1,\dots,N+3}$ is the cartesian basis of $\mathbb{R}^{N+3}$ and $\{u_k\}_{k=1,\dots,N+2}$ that of $\mathbb{R}^{N+2}$. The remaining $\mathbf{x}_k$ are defined as

$$\mathbf{x}_k = \mathbf{x}_0 - \sum_{i=0}^{k-1} \gamma_{k,i}\mathbf{g}_i, \quad \text{for } k = 1,\dots,N.$$

Finally, this allows expressing (3.17) as the following linear semidefinite problem

$$\begin{aligned} C(N,R) = \sup_{\substack{F\in\mathbb{R}^{1\times N+2}\\ G\in\mathbf{S}^+_{N+3}}} & \operatorname{Tr}(A_{\text{obj}}G) + Fa_{\text{obj}} \\ \text{s.t. } & F\mathbf{f}_j \geq F\mathbf{f}_i + \mathbf{g}_i^T G(\mathbf{x}_j - \mathbf{x}_i) \\ & \quad + \tfrac{1}{2(L-\mu)}(\mathbf{g}_i - \mathbf{g}_j - \mu(\mathbf{x}_i - \mathbf{x}_j))^T G(\mathbf{g}_i - \mathbf{g}_j - \mu(\mathbf{x}_i - \mathbf{x}_j)) \\ & \quad + \tfrac{\mu}{2}(\mathbf{x}_i - \mathbf{x}_j)^T G(\mathbf{x}_i - \mathbf{x}_j), \text{ for } i,j \in \{*,0,\dots,N\}, \\ & R^2 \geq \operatorname{Tr}(A_{\text{init}}G) + Fa_{\text{init}}. \end{aligned} \tag{3.19}$$

This problem can be solved efficiently using semidefinite solvers (e.g., Mosek [Mosek, 2010], SDPT3 [Toh et al., 2012] or Sedumi [Sturm, 1999]).

**Remark 3.2.2.** *We can show using a simple homogeneity argument inside* (3.19) *that*

$$C(N,R) = C(N,1)R^2,$$

*and therefore, we can set* $R = 1$ *without loss of generality.*

**Remark 3.2.3.** *In many situations, PEPs admit the ratio* $\Phi_{d,\text{obj}}(S)/\Phi_{d,\text{init}}(S)$ *as objective, without the constraint* $\Phi_{d,\text{init}}(S) \leq R^2$. *Note that this can be reformulated equivalently as* (3.19) *with* $R = 1$ *using again homogeneity arguments. This is not true in Chapter 5, as we consider objective criteria and initial conditions with additional constant terms, breaking the homogeneity argument.*

**Example: Gradient descent with constant step size** We illustrate the previous developments on one of the most celebrated (and simple) optimization method which is the gradient descent with constant step size. Given a function $f \in \mathcal{F}_{0,\text{L}}(\mathbb{R}^d)$ ($\mu = 0$), a starting point $x_0 \in \mathbb{R}^d$ and a step size $\lambda \geq 0$ gradient descent updates are expressed as

$$x_{k+1} = x_k - \lambda\nabla f(x_k), \text{ for } k \geq 0.$$

This method can be formulated as (3.13) by setting the coefficients $\gamma_{i,j}$ all equal to $\lambda$. We study the worst-case guarantees of the objective criterion $\Phi_{d,\text{obj}}(f,\{x_k\}_{k=0,\dots,N}) = f(x_N) - f(x_*)$ with $x_* \in \operatorname{argmin}_x f(x)$ and with initial condition $\Phi_{d,\text{init}}(f,\{x_k\}_{k=0,\dots,N}) = \|x_0 - x_*\|^2$ (see e.g. Abbaszadehpeivasti et al. [2021] for worst-case guarantees on gradient descent in a different setting). Note that these quantities satisfy Definition 3.2.1. Numerical computations of the bound $C(N,1)$ on $f(x_N) - f(x_*)$ with initial condition $\|x_0 - x_*\|^2 \leq 1$ are represented in Figure 3.2 (see §Codes for a matlab script reproducing this figure).

Feasible points to (3.19) provide lower bounds on $C(N,R)$ (i.e. lower bounds on the worst-behavior of a method). In order to obtain upper bounds, it is natural to study dual problems associated to (3.19). Before studying a dual formulation, let us show that Slater condition (see e.g., Boyd and Vandenberghe [2004, Section 5.2.3]) holds for (3.19), which imply zero duality gap (i.e. the largest lower bound is the same as the smallest upper bound).

**Slater condition** We start by observing that all the constraints in (3.19) are affine and therefore, Slater condition simply corresponds to find elements $G \in \mathbf{S}^+_{N+3}$ and $F \in \mathbb{R}^{1\times N+2}$ such that

$$G \succ 0,$$

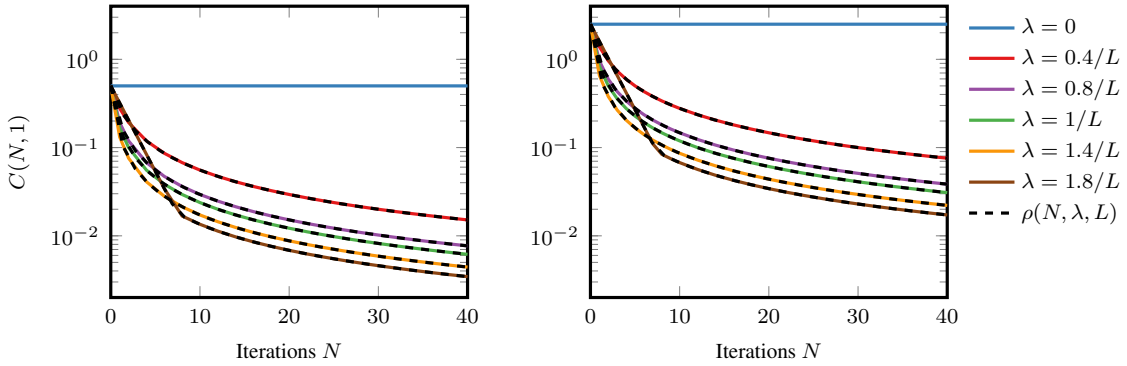


Figure 3.2: Numerical worst-case guarantees on $f(x_N) - f(x_*)$ with initial condition $\|x_0 - x_*\|^2 \leq 1$, as function of $N$ (obtained by solving semidefinite programs (3.19)) with $L = 1$ and $R = 1$. The semidefinite programs were solved through Löfberg [2004] and Mosek [2010]. The dashed black curves corresponds to the conjectured upper bound $\rho(N,\lambda,L) = \frac{L}{2}\max\left((1-L\lambda)^{2N}, \frac{1}{2NL\lambda+1}\right)$ from Drori and Teboulle [2014, Conjecture 3.1].

$$\begin{aligned} F\mathbf{f}_j \geq & F\mathbf{f}_i + \mathbf{g}_i^T G(\mathbf{x}_j - \mathbf{x}_i) + \tfrac{1}{2(L-\mu)}(\mathbf{g}_i - \mathbf{g}_j - \mu(\mathbf{x}_i - \mathbf{x}_j))^T G(\mathbf{g}_i - \mathbf{g}_j - \mu(\mathbf{x}_i - \mathbf{x}_j)) \\ & + \tfrac{\mu}{2}(\mathbf{x}_i - \mathbf{x}_j)^T G(\mathbf{x}_i - \mathbf{x}_j), \text{ for } i,j \in \{*,0,\dots,N\}, \end{aligned} \tag{3.20}$$

and

$$R^2 \geq \mathrm{Tr}(A_{\text{init}} G) + F a_{\text{init}}. \tag{3.21}$$

The two first constraints can be satisfied by finding some integer $d > 0$, a starting point $x_0 \in \mathbb{R}^d$ and a function $f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$ such that the Gram matrix $G$ defined as in (3.18) has full rank. After that, one can satisfy easily the last constraint using an homogeneity argument.

We denote by $f$, the quadratic function define as

$$f(x) = \tfrac{1}{2}(x - x_*)^T \left[H + \mu I\right](x - x_*),$$

where $x_* = e_{N+3}$ ($\{e_k\}_{k=1,\dots,N+3}$ is the cartesian basis of $\mathbb{R}^{N+3}$) and $A \in \mathbf{S}_{N+3}$ is the symmetric tridiagonal matrix

$$H = \tfrac{L-\mu}{2\cos(\frac{\pi}{N+4})} \begin{pmatrix} \cos(\frac{\pi}{N+4}) & 1/2 & 0 & \dots & \dots & \dots & 0 \\ 1/2 & \cos(\frac{\pi}{N+4}) & 1/2 & 0 & \dots & \dots & 0 \\ 0 & 1/2 & \cos(\frac{\pi}{N+4}) & 1/2 & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \end{pmatrix}.$$

From e.g. Bernstein [2009, Fact 5.11.43] we obtain that $H$ satisfies $\mathrm{Sp}(H) = \{\lambda_k\}_{k=1,\dots,N+3}$ with

$$\lambda_k = \tfrac{L-\mu}{2\cos(\frac{\pi}{N+4})}(\cos(\tfrac{k\pi}{N+4}) + \cos(\tfrac{\pi}{N+4})), \quad \text{for } k = 1,\dots,N+4.$$

In particular, we observe that $L - \mu = \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{N+3} = 0$, and therefore $f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^{N+3})$.

Finally, we chose $x_0 = e_1 + e_{N+3}$ and set the iterates $\{x_k\}_{k=1,\dots,N}$ as

$$x_k = x_0 - \sum_{i=0}^{k-1} \gamma_{k,i} \nabla f(x_i), \text{ for } k = 1,\dots,N,$$

where $\{\gamma_{i,j}\}_{i,j}$ is a predetermined sequence of parameters with $\gamma_{k,k-1} \neq 0$ for $k = 1, \ldots, N$.

Starting from the observation that $\nabla f(x_0) = [H + \mu I]\,(x_0 - x_*) \in \mathrm{span}[e_1, e_2] \backslash \mathrm{span}[e_1]$, we can show recursively using the tridiagonal form of $[H + \mu I]$ as well as $\gamma_{k,k-1} \neq 0$ for $k = 1, \ldots, N$, that

$$\nabla f(x_k) \in \mathrm{span}[e_1, \ldots, e_{k+2}] \backslash \mathrm{span}[e_1, \ldots, e_{k+1}], \text{ for } k = 0, \ldots, N.$$

This implies that the matrix

$$X = [x_*, x_0, \nabla f(x_0), \ldots, \nabla f(x_N)] \in \mathbb{R}^{N+3 \times N+3},$$

is full rank and therefore,

$$G \succ 0.$$

In addition, Theorem 3.1.14 ensures that (3.20) is satisfied. If (3.21) is not satisfied, it is sufficient to divide $F$ and $G$ by $\mathrm{Tr}(A_{\text{init}} G) + F a_{\text{init}} > R^2 > 0$.

Hence, Slater condition holds for nondegenerate fixed-step gradient methods on the class of smooth closed strongly convex and proper functions, and therefore strong duality holds for (3.19).

### 3.2.2 Dual formulation

We denote by $\nu = \{\nu_{i,j}\}_{i,j \in \{*,0,\ldots,N\}}$ the nonnegative Lagrange multipliers associated with the convex interpolation inequalities and $\tau$ that associated with the initial condition. We define respectively $\tilde{G}(\nu) \in \mathbb{R}^{N+3 \times N+3}$ and $\tilde{F}(\nu) \in \mathbb{R}^{N+2}$ such that

$$\begin{aligned} \tilde{G}(\nu) = \sum_{i,j \in \{*,0,\ldots,N\}} \nu_{i,j} \big[ &(\mathbf{x}_j - \mathbf{x}_i)\mathbf{g}_i^T + \tfrac{\mu}{2}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \\ &+ \tfrac{1}{2(L-\mu)}(\mathbf{g}_i - \mathbf{g}_j - \mu(\mathbf{x}_i - \mathbf{x}_j))(\mathbf{g}_i - \mathbf{g}_j - \mu(\mathbf{x}_i - \mathbf{x}_j))^T \big], \end{aligned} \tag{3.22}$$

and

$$\tilde{F}(\nu) = \sum_{i,j \in \{*,0,\ldots,N\}} \nu_{i,j}(\mathbf{f}_i - \mathbf{f}_j). \tag{3.23}$$

The Lagrangian associated with (3.19) is defined as

$$\mathcal{L}(G, F, \nu, \tau) = F\left[a_{\text{obj}} - \tau a_{\text{init}} - \tilde{F}(\nu)\right] + \mathrm{Tr}\left(\left[A_{\text{obj}} - \tau A_{\text{init}} - \tilde{G}(\nu)\right] G\right) + \tau R^2, \tag{3.24}$$

and the corresponding dual function is

$$\sup_{G \succcurlyeq 0, F} \mathcal{L}(G, F, \nu, \tau) = \begin{cases} \tau R^2 & \text{if } a_{\text{obj}} - \tau a_{\text{init}} - \tilde{F}(\nu) = 0 \text{ and} \\ & \quad A_{\text{obj}} - \tau A_{\text{init}} - \tilde{G}(\nu) + \left(A_{\text{obj}} - \tau A_{\text{init}} - \tilde{G}(\nu)\right)^T \preccurlyeq 0 \\ +\infty & \text{otherwise,} \end{cases} \tag{3.25}$$

leading to the dual problem

$$\begin{aligned} &\inf_{\substack{\nu \geq 0 \\ \tau \geq 0}} \tau R^2 \\ &\text{s.t. } a_{\text{obj}} - \tau a_{\text{init}} - \tilde{F}(\nu) = 0, \\ &\quad A_{\text{obj}} - \tau A_{\text{init}} - \tilde{G}(\nu) + \left(A_{\text{obj}} - \tau A_{\text{init}} - \tilde{G}(\nu)\right)^T \preccurlyeq 0, \end{aligned} \tag{3.26}$$

which is also a linear semidefinite program.

We have seen previously in Section 3.2.1 that strong duality holds. Therefore, given a nondegenerate first-order method $\mathcal{M}$ (as in 3.13), Gram-representable objective criterion $\Phi_{d,\text{obj}}$ and initial condition $\Phi_{d,\text{init}}$, we obtain that $C(N,R)$ the smallest quantity such that

$$\begin{aligned}&\forall d \in \mathbb{N}^*,\ \forall f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d),\ \forall x_0 \in \mathbb{R}^d,\ \text{s.t. } \Phi_{d,\text{init}}(f,\{x_k\}_{k=0,\dots,N}) \leq \mathbb{R}^2\\ &\Phi_{d,\text{obj}}(f,\{x_k\}_{k=0,\dots,N}) \leq C(N,R),\end{aligned} \tag{3.27}$$

with

$$x_k = \mathcal{M}(f,x_0,k),\ \text{for } k = 1,\dots,N,$$

is equal to $\tau_* R^2$ with $(\tau_*,\nu_*)$ an optimal solution to the dual problem (3.26). Hence, we recover the result of Remark 3.2.2, stating that $C(N,R)$ is 1-homogeneous in $R^2$.

Moreover, accessing dual feasible pairs $(\nu,\tau)$ allows writing proofs of worst-case bounds using weighted combinations of inequalities. Indeed, let $(\nu,\tau)$ be a feasible dual pair, then for all $G \succcurlyeq 0$ and $F$, we can write

$$0 \geq F\left[a_{\text{obj}} - \tau a_{\text{init}} - \tilde{F}(\nu)\right] + \mathrm{Tr}\left(\left[A_{\text{obj}} - \tau A_{\text{init}} - \tilde{G}(\nu)\right] G\right).$$

Using functional notations with $f$, it transcribes as $\forall d \in \mathbb{N}^*,\ \forall f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$

$$\begin{aligned}0 \geq\ &\Phi_{d,\text{obj}}(f,\{x_k\}_{k=0,\dots,N}) - \tau\Phi_{d,\text{init}}(f,\{x_k\}_{k=0,\dots,N})\\ &- \sum_{i,j\in\{*,0,\dots,N\}} \nu_{i,j}\big[f(x_i) - f(x_j) + \langle \nabla f(x_i), x_j - x_i\rangle + \tfrac{\mu}{2}\|x_i - x_j\|^2\\ &\qquad + \tfrac{1}{2(L-\mu)}\|\nabla f(x_i) - \nabla f(x_j) - \mu(x_i - x_j)\|^2\big]\\ \geq\ &\Phi_{d,\text{obj}}(f,\{x_k\}_{k=0,\dots,N}) - \tau\Phi_{d,\text{init}}(f,\{x_k\}_{k=0,\dots,N}),\end{aligned}$$

where the last inequality comes from the fact that the $\nu_{i,j}$ are nonnegative and the terms that multiply it are nonpositive using e.g. Theorem 3.1.17. Many proofs of worst-case guarantees in different chapters of this manuscript were obtained following the similar procedures, that is

(i) intuit analytical expressions for $\tau$ and $\nu$. This step is often an "educated guess" as we infer these expressions using numerical solutions to (3.26) with different values of $N$, $\mu$ and $L$, together with a computer algebra system.

(ii) Compute

$$\begin{aligned}&\Phi_{d,\text{obj}}(f,\{x_k\}_{k=0,\dots,N}) - \tau\Phi_{d,\text{init}}(f,\{x_k\}_{k=0,\dots,N})\\ &- \sum_{i,j\in\{*,0,\dots,N\}} \nu_{i,j}\big[f(x_i) - f(x_j) + \langle \nabla f(x_i), x_j - x_i\rangle + \tfrac{\mu}{2}\|x_i - x_j\|^2\\ &\qquad + \tfrac{1}{2(L-\mu)}\|\nabla f(x_i) - \nabla f(x_j) - \mu(x_i - x_j)\|^2\big]\end{aligned}$$

and show that it is nonpositive (in particular that it can be written as a nonpositive sum of squared euclidean norms).

(iii) Conclude that

$$\Phi_{d,\text{obj}}(f,\{x_k\}_{k=0,\dots,N}) \leq \tau\Phi_{d,\text{init}}(f,\{x_k\}_{k=0,\dots,N}),$$

for all $d \in \mathbb{N}^*$, all functions in the considered functional class ($\mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$ in that case) and all initial point $x_0 \in \mathbb{R}^d$.

If we obtain in step (i), $\tau = \tau_* = C(N,1)$ an optimal dual point, we can conclude in step (iii) that

$$\Phi_{d,\text{obj}}(f, \{x_k\}_{k=0,\dots,N}) \leq C(N,1)\Phi_{d,\text{init}}(f, \{x_k\}_{k=0,\dots,N}),$$

and there exists a dimension $d$ and function $f \in \mathcal{F}_{\mu,\text{L}}(\mathbb{R}^d)$ such that

$$\Phi_{d,\text{obj}}(f, \{x_k\}_{k=0,\dots,N}) = C(N,1)\Phi_{d,\text{init}}(f, \{x_k\}_{k=0,\dots,N}).$$

**Remark 3.2.4.** *One could use other inequalities satisfied by elements of $\mathcal{F}_{\mu,\text{L}}(\mathbb{R}^d)$, for instance we could replace some of the interpolation inequalities by the (perhaps more standard) inequalities*

$$f_j \leq f_i + \langle g_i, x_j - x_i\rangle + \tfrac{L}{2}\|x_i - x_j\|^2$$
*and/or*
$$f_j \geq f_i + \langle g_i, x_j - x_i\rangle + \tfrac{\mu}{2}\|x_i - x_j\|^2,$$

*for some $i, j \in \{*, 0, \dots, N\}$. This would only produce an upper bound on $C(N,R)$ but could induce simpler analytical expressions for the dual variables as well as simpler proofs.*

### 3.2.3 Performance estimation with Lyapunov functions

We end this introductory chapter on performance estimation by presenting useful techniques that allow using and designing Lyapunov functions (i.e., objective criterion and initial condition) in the performance estimation framework. These techniques or related ones have been used to obtain some results of Chapters 4 to 6.

Originating from the study of dynamical systems, Lyapunov's theory [Lyapunov and Fuller, 1992] aims at finding a sequence of functions (called potential functions or Lyapunov functions) that are conserved or dissipated (i.e decreasing) with time. Using conservation or dissipation together with the form of the potentials can be used to provide convergence guarantees for the system toward some stationary state. This approach has been applied for optimization algorithms with e.g. Nesterov [1983], Bansal and Gupta [2019], Wilson [2018], Wilson et al. [2021] and incorporated in PEPs by Taylor and Bach [2019], Taylor et al. [2018b].

In this section, we look at gradient methods that exhibit some recursive formulation, that is methods satisfying (3.13) and for some $m \in \mathbb{N}^*$, $x_0 \in \mathbb{R}^d$,

$$\begin{cases} x_k &= x_0 - \sum_{i=0}^{k-1} \gamma_{k,i}\nabla f(x_i), \text{ for } k = 1, \dots, m-1 \\ x_{k+m} &= \sum_{i=0}^{m-1} \left[\alpha_{k+m,i}x_{k+i} - \gamma_{k+m,i}\nabla f(x_{k+i})\right], \text{ for } k \geq 0 \end{cases} \tag{3.28}$$

with $\sum_{i=0}^{m-1} \alpha_{k+m,i} = 1$ (in order for $x_*$ satisfying $\nabla f(x_*) = 0$ to be a stationary point of the method), and $a_{k+m,m-1} \neq 0$ (i.e., nondegenerate methods).

Finding potentials functions for a given method corresponds to obtain a sequence of functions $\{\Phi_k\}_{k\in\mathbb{N}}$ satisfying

$$\Phi_{N+1}(f, \{x_k\}_{k=N+1,\dots,N+m}) \leq \Phi_N(f, \{x_k\}_{k=N,\dots,N+m-1}), \tag{3.29}$$

for all $d \in \mathbb{N}^*$, all $f \in \mathcal{F}_{\mu,\text{L}}(\mathbb{R}^d)$, and all $N \in \mathbb{N}$. This allows nesting these inequalities together and producing the following bound after $N$ iterations

$$\Phi_N(f, \{x_k\}_{k=N,\dots,N+m-1}) \leq \Phi_0(f, \{x_k\}_{k=0,\dots,m-1}), \tag{3.30}$$

for all $d \in \mathbb{N}^*$, all $f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$, and all $N \in \mathbb{N}$.

When the potential functions are Gram-representable (see Definition 3.2.1), we can use the performance estimation methodology to check whether (3.29) is satisfied. Indeed, this corresponds to verify that

$$\begin{aligned} 0 \geq \sup_{d,\, S}\ & \Phi_{N+1}(S) - \Phi_N(S) \\ \text{s.t. } & S = \{(x_k, g_k, f_k)\}_{k\in\{*,N,\dots,N+m\}}, \\ & f_j \geq f_i + \langle g_i, x_j - x_i\rangle + \tfrac{1}{2(L-\mu)}\|g_i - g_j - \mu(x_i - x_j)\|^2 \\ & \qquad + \tfrac{\mu}{2}\|x_i - x_j\|^2 \text{ for } i,j \in \{*, N, \dots, N+m\}, \\ & g_* = 0, \\ & x_{N+m} = \sum_{i=0}^{m-1} \left[\alpha_{N+m,i} x_{N+i} - \gamma_{N+m,i}\nabla f(x_{N+i})\right], \end{aligned} \tag{3.31}$$

for all $N \geq 0$.

**Remark 3.2.5.** *We see that (3.31) does not take into account how iterates $x_N, \dots, x_{N+m-1}$ were obtained. This makes the worst-case guarantees obtained using this potential based formulation more conservative than the ones presented in the previous section, but often leads to simpler analyses.*

We can proceed as in Section 3.2.1 and write (3.31) as

$$\begin{aligned} 0 \geq \sup_{\substack{F\in\mathbb{R}^{m+2}\\ G\in\mathbf{S}^+_{2m+2}}}\ & \mathrm{Tr}([A_{N+1} - A_N]\, G) + F\,[a_{N+1} - a_N] \\ \text{s.t. } & F\mathbf{f}_j \geq F\mathbf{f}_i + \mathbf{g}_i^T G(\mathbf{x}_j - \mathbf{x}_i) \\ & \qquad + \tfrac{1}{2(L-\mu)}(\mathbf{g}_i - \mathbf{g}_j - \mu(\mathbf{x}_i - \mathbf{x}_j))^T G(\mathbf{g}_i - \mathbf{g}_j - \mu(\mathbf{x}_i - \mathbf{x}_j)) \\ & \qquad + \tfrac{\mu}{2}(\mathbf{x}_i - \mathbf{x}_j)^T G(\mathbf{x}_i - \mathbf{x}_j), \text{ for } i,j \in \{*, N, \dots, N+m\}, \end{aligned} \tag{3.32}$$

where

$$\begin{aligned} & \mathbf{x}_* = e_1,\ \mathbf{g}_* = 0,\ \mathbf{f}_* = u_1, \\ & \mathbf{x}_k = e_{k-N+2} \text{ for } k = N, \dots, N+m-1,\ \mathbf{g}_k = e_{m+k-N+2} \text{ for } k = N, \dots, N+m, \\ & \mathbf{x}_{N+m} = \sum_{i=0}^{m-1} \left[\alpha_{N+m,i}\mathbf{x}_{N+i} - \gamma_{N+m,i}\mathbf{g}_{N+i}\right] \\ & \mathbf{f}_k = u_{k-N+2} \text{ for } k = N, \dots, N+m, \end{aligned}$$

and the potentials satisfy

$$\Phi_k(S) = \Phi_k(F, G) = \mathrm{Tr}(A_k G) + F a_k, \text{ for } k \geq 0.$$

Using similar duality arguments as in Section 3.2.2, we can show that verifying (3.32) amounts to solve the feasibility problem

$$\begin{aligned} \text{Find } & \{\nu_{i,j}\}_{i,j\in\{*,N,\dots,N+m\}} \\ \text{s.t. } & \nu \geq 0 \\ & a_{N+1} - a_N - \tilde{F}(\nu) = 0, \\ & A_{N+1} - a_N - \tilde{G}(\nu) + \left(A_{N+1} - A_N - \tilde{G}(\nu)\right)^T \preccurlyeq 0, \end{aligned} \tag{3.33}$$

with

$$\tilde{F}(\nu) = \sum_{i,j \in \{*,N,\dots,N+m\}} \nu_{i,j} \left[\mathbf{f}_i - \mathbf{f}_j\right],$$

and

$$\tilde{G}(\nu) = \sum_{i,j \in \{*,N,\dots,N+m\}} \nu_{i,j} \Big[(\mathbf{x}_j - \mathbf{x}_i)\mathbf{g}_i^T + \tfrac{\mu}{2}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T + \tfrac{1}{2(L-\mu)}(\mathbf{g}_i - \mathbf{g}_j - \mu(\mathbf{x}_i - \mathbf{x}_j))(\mathbf{g}_i - \mathbf{g}_j - \mu(\mathbf{x}_i - \mathbf{x}_j))^T\Big].$$

Therefore, if (3.33) is feasible, (3.29) holds as well as (3.30).

In the following, we illustrate how introducing some degrees of freedom in the parameters $A_k$'s and $a_k$'s (i.e. in the potential functions) allows obtaining simple convergence arguments for two optimization methods.

**Example: Nesterov method with constant momentum** In this part, we study the standard Nesterov method with constant momentum [Nesterov, 2018, Algorithm 2.2.22] and design a potential function to show linear convergence of the algorithm with the accelerated rate $\rho = \left(1 - \sqrt{\mu/L}\right)$. Potential based proof of many accelerated methods can be found in e.g. in Wilson et al. [2021] and an analysis of Nesterov's method with constant momentum based on linear matrix inequality (originating from control theory, and essentially corresponding to relaxed version of PEPs) is performed in Hu and Lessard [2017].

When applied to smooth strongly convex functions $f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$, starting from $y_0 = x_0 \in \mathbb{R}^d$ the updates are often expressed using 2 sequences as

$$\left\{ \begin{array}{rcl} y_{k+1} & = & x_k - \frac{1}{L}\nabla f(x_k) \\ x_{k+1} & = & y_{k+1} + \beta(y_{k+1} - y_k), \text{ for } k \geq 0, \end{array} \right. \tag{3.34}$$

with $\beta = \left(\sqrt{L} - \sqrt{\mu}\right) / \left(\sqrt{L} + \sqrt{\mu}\right)$. Note that the sequences names are inverted compared to Nesterov [2018, Algorithm 2.2.22], mainly for notational coherence. This method can be expressed under the form (3.28) with $m = 2$ as

$$\left\{ \begin{array}{rcl} x_1 & = & x_0 - \frac{1}{L}\nabla f(x_0) \\ x_{k+2} & = & (1+\beta)x_{k+1} - \beta x_k - \frac{1+\beta}{L}\nabla f(x_{k+1}) + \frac{\beta}{L}\nabla f(x_k), \text{ for } k \geq 0, \end{array} \right.$$

and we focus on a particular class of potentials functions defined as

$$\Phi_k(F,G) = \frac{1}{\rho^k}\left(F\underbrace{(\mathbf{f}_k - \mathbf{f}_*)}_{a_k} + \mathrm{Tr}\left(\underbrace{[\mathbf{x}_{k+1} - \mathbf{x}_*, \mathbf{x}_k - \mathbf{x}_*, \frac{\mathbf{g}_k}{L}]S[\mathbf{x}_{k+1} - \mathbf{x}_*, \mathbf{x}_k - \mathbf{x}_*, \frac{\mathbf{g}_k}{L}]^T}_{A_k} G\right)\right),$$

for $k \geq 0$ with $S \in \mathbf{S}_3^+$. Similar to (3.33), we can obtain a feasibility problem ensuring that (3.29) is satisfied. In addition, we can add the matrix $S$ together with its semidefinite constraint in the feasibility

problem as

$$\begin{aligned}
&\text{find } \{\nu_{i,j}\}_{i,j\in\{*,N,\dots,N+2\}} \text{ and } S \in \mathbb{R}^{3\times 3}\\
&\text{s.t. } \nu \geq 0,\ S \succcurlyeq 0,\\
&\quad \mathbf{f}_{N+1} - \mathbf{f}_* - \rho(\mathbf{f}_N - \mathbf{f}_*) - \tilde{F}(\nu) = 0,\\
&\quad [\mathbf{x}_{N+2} - \mathbf{x}_*, \mathbf{x}_{N+1} - \mathbf{x}_*, \mathbf{g}_{N+1}/L] S [\mathbf{x}_{N+2} - \mathbf{x}_*, \mathbf{x}_{N+1} - \mathbf{x}_*, \mathbf{g}_{N+1}/L]^T\\
&\quad - \rho[\mathbf{x}_{N+1} - \mathbf{x}_*, \mathbf{x}_N - \mathbf{x}_*, \mathbf{g}_N/L] S [\mathbf{x}_{N+1} - \mathbf{x}_*, \mathbf{x}_N - \mathbf{x}_*, \mathbf{g}_N/L]^T - \tfrac{\tilde{G}(\nu)+\tilde{G}(\nu)^T}{2} \preccurlyeq 0.
\end{aligned} \tag{3.35}$$

We can notice that feasibility of (3.35) is independent of $N$ in that case, and therefore we can study this problem with $N = 0$ without loss of generality.

Problem (3.35) is a linear semidefinite program and can be solved efficiently as previously (see §Codes for a matlab script solving this problem). We typically search for feasible points with sparse multiplier $\nu$ and low rank potential matrix $S$. In this example, numerical trials indicate that

$$S = \frac{L^2}{2(L-\mu)} \begin{pmatrix} \left(1+\sqrt{\frac{\mu}{L}}\right)^2 & -\left(1+\sqrt{\frac{\mu}{L}}\right) & \left(1+\sqrt{\frac{\mu}{L}}\right) \\ -\left(1+\sqrt{\frac{\mu}{L}}\right) & 1 & -1 \\ \left(1+\sqrt{\frac{\mu}{L}}\right) & -1 & 1 \end{pmatrix}$$

and

$$\nu_{N+1,*} = (1-\rho),\ \nu_{N+1,N} = \rho \text{ and } \nu_{i,j} = 0 \text{ otherwise},$$

are feasible. Note that the potential that we obtained is slightly different from that of Hu and Lessard [2017, Section 3.1] or Bansal and Gupta [2019, Section 5.5] (essentially as we consider function evaluations on the extrapolated sequence). We can then combine interpolation inequalities as presented in Section 3.2.2 to obtain the following property.

**Proposition 3.2.6.** *Let $d \in \mathbb{N}^*$, $0 < \mu < L < +\infty$, $f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$ and $x_N, x_{N+1} \in \mathbb{R}^d$ be some iterates. It holds that*

$$\Phi(x_{N+2}, x_{N+1}) \leq \rho \Phi(x_{N+1}, x_N),$$

*where*

$$\Phi(x,y) = \tfrac{L^2}{2(L-\mu)} \|(1+\sqrt{\tfrac{\mu}{L}})(x-x_*) - (y-x_*) + \tfrac{1}{L}\nabla f(y)\|^2 + f(y) - f(x_*),$$

*$x_* \in \operatorname{argmin}_x f(x)$, $\rho = \left(1-\sqrt{\mu/L}\right)$, and $x_{N+2}$ is constructed using Nesterov's method as*

$$x_{N+2} = (1+\beta)x_{N+1} - \beta x_N - \tfrac{1+\beta}{L}\nabla f(x_{N+1}) + \tfrac{\beta}{L}\nabla f(x_N),$$

*with $\beta = \left(\sqrt{L}-\sqrt{\mu}\right) \Big/ \left(\sqrt{L}+\sqrt{\mu}\right)$.*

*Proof.* The proof consists in performing the following weighted sum of inequalities

- smoothness and strong convexity of $f$ between $x_{N+1}$ and $x_*$ with weight $\nu_{N+1,*} = 1-\rho$,

$$\begin{aligned} f(x_*) \geq f(x_{N+1}) + \langle \nabla f(x_{N+1}), x_* - x_{N+1}\rangle + \tfrac{\mu}{2}\|x_{N+1} - x_*\|^2 \\ + \tfrac{1}{2(L-\mu)}\|\nabla f(x_{N+1}) - \mu(x_{N+1} - x_*)\|^2, \end{aligned}$$

- smoothness and strong convexity of $f$ between $x_{N+1}$ and $x_N$ with weight $\nu_{N+1,N} = \rho$,

$$f(x_N) \geq f(x_{N+1}) + \langle \nabla f(x_{N+1}), x_N - x_{N+1} \rangle + \tfrac{\mu}{2}\|x_{N+1} - x_N\|^2 \\ + \tfrac{1}{2(L-\mu)}\|\nabla f(x_{N+1}) - \nabla f(x_N) - \mu(x_{N+1} - x_N)\|^2.$$

This weighted sum writes as

$$\begin{aligned}(1-\rho)f(x_*) + \rho f(x_N) \geq & f(x_{N+1}) + (1-\rho)\big[\langle \nabla f(x_{N+1}), x_* - x_{N+1}\rangle + \tfrac{\mu}{2}\|x_{N+1} - x_*\|^2 \\ & + \tfrac{1}{2(L-\mu)}\|\nabla f(x_{N+1}) - \mu(x_{N+1} - x_*)\|^2\big] \\ & + \rho\big[\langle \nabla f(x_{N+1}), x_N - x_{N+1}\rangle + \tfrac{\mu}{2}\|x_{N+1} - x_N\|^2 \\ & + \tfrac{1}{2(L-\mu)}\|\nabla f(x_{N+1}) - \nabla f(x_N) - \mu(x_{N+1} - x_N)\|^2\big].\end{aligned} \tag{3.36}$$

The terms depending on $\nabla f(x_{k+1})$ in this inequality can be grouped together as

$$\begin{aligned}&\langle \nabla f(x_{k+1}), \tfrac{(1-\rho)L}{L-\mu}(x_* - x_{N+1}) + \tfrac{\rho L}{L-\mu}(x_N - x_{N+1}) - \tfrac{\rho}{L-\mu}\nabla f(x_N)\rangle + \tfrac{1}{2(L-\mu)}\|\nabla f(x_{N+1})\|^2 \\ =&\tfrac{L^2}{2(L-\mu)}\|\tfrac{1}{L}\nabla f(x_{N+1}) + \rho(x_N - x_*) - (x_{N+1} - x_*) - \tfrac{\rho}{L}\nabla f(x_N)\|^2 \\ &- \tfrac{L^2}{2(L-\mu)}\|\rho(x_N - x_*) - (x_{N+1} - x_*) - \tfrac{\rho}{L}\nabla f(x_N)\|^2 \\ =&\tfrac{L^2}{2(L-\mu)}\|(2-\rho)(x_{N+2} - x_*) - (x_{N+1} - x_*) + \tfrac{1}{L}\nabla f(x_{N+1})\|^2 \\ &- \tfrac{L^2}{2(L-\mu)}\|\rho(x_N - x_*) - (x_{N+1} - x_*) - \tfrac{\rho}{L}\nabla f(x_N)\|^2,\end{aligned}$$

where in the last equality we used $x_{N+2} = (1+\beta)x_{N+1} - \beta x_N - \frac{1+\beta}{L}\nabla f(x_{N+1}) + \frac{\beta}{L}\nabla f(x_N)$ and $\beta = \rho/(2-\rho)$. We inject this expression in (3.36) and get

$$\begin{aligned}0 \geq &\Phi(x_{N+2}, x_{N+1}) - \rho(f(x_N) - f(x_*)) + \tfrac{(1-\rho)L\mu}{2(L-\mu)}\|x_{N+1} - x_*\|^2 + \tfrac{\rho L\mu}{2(L-\mu)}\|x_{N+1} - x_N\|^2 \\ &+ \tfrac{\rho\mu}{L-\mu}\langle \nabla f(x_N), x_{N+1} - x_N\rangle + \tfrac{\rho}{2(L-\mu)}\|\nabla f(x_N)\|^2 \\ &- \tfrac{L^2}{2(L-\mu)}\|\rho(x_N - x_*) - (x_{N+1} - x_*) - \tfrac{\rho}{L}\nabla f(x_N)\|^2,\end{aligned} \tag{3.37}$$

Introducing $\Phi(x_{N+1}, x_N)$ in (3.37) allows writing

$$\begin{aligned}0 \geq &\Phi(x_{N+2}, x_{N+1}) - \rho\Phi(x_{N+1}, x_N) + \tfrac{(1-\rho)L\mu}{2(L-\mu)}\|x_{N+1} - x_*\|^2 + \tfrac{\rho L\mu}{2(L-\mu)}\|x_{N+1} - x_N\|^2 \\ &+ \tfrac{\rho L^2}{2(L-\mu)}\|(2-\rho)(x_{N+1} - x_*) - (x_N - x_*) + \tfrac{1}{L}\nabla f(x_N)\|^2 \\ &+ \tfrac{\rho\mu}{L-\mu}\langle \nabla f(x_N), x_{N+1} - x_N\rangle + \tfrac{\rho}{2(L-\mu)}\|\nabla f(x_N)\|^2 \\ &- \tfrac{L^2}{2(L-\mu)}\|\rho(x_N - x_*) - (x_{N+1} - x_*) - \tfrac{\rho}{L}\nabla f(x_N)\|^2.\end{aligned} \tag{3.38}$$

As previously, we group together the terms in (3.38) ($\Phi(x_{N+2}, x_{N+1})$ and $\rho\Phi(x_{N+1}, x_N)$ apart) that depend on $x_{N+1} - x_*$, as

$$\begin{aligned}&L\tfrac{\rho(2-\rho)^2 L + \mu - L}{2(L-\mu)}\|x_{N+1} - x_*\|^2 - L\tfrac{\rho L(2-\rho) + \rho\mu - L\rho}{(L-\mu)}\langle x_{N+1} - x_*, x_N - x_* - \tfrac{1}{L}\nabla f(x_N)\rangle \\ =&\tfrac{L}{2}\sqrt{\tfrac{\mu}{L}}\|x_{N+1} - x_*\|^2 - L\sqrt{\tfrac{\mu}{L}}\langle x_{N+1} - x_*, x_N - x_* - \tfrac{1}{L}\nabla f(x_N)\rangle \\ =&\tfrac{L}{2}\sqrt{\tfrac{\mu}{L}}\|x_{N+1} - x_N + \tfrac{1}{L}\nabla f(x_N)\|^2 - \tfrac{L}{2}\sqrt{\tfrac{\mu}{L}}\|x_N - x_* - \tfrac{1}{L}\nabla f(x_N)\|^2,\end{aligned}$$

where we used $\rho = 1 - \sqrt{\mu/L}$ in the first equality. Re-injecting this expression in (3.38) leads to

$$
\begin{aligned}
0 \geq & \Phi(x_{N+2}, x_{N+1}) - \rho\Phi(x_{N+1}, x_N) + \tfrac{L}{2}(1-\rho)\|x_{N+1} - x_N + \tfrac{1}{L}\nabla f(x_N)\|^2 \\
& - \tfrac{L}{2}(1-\rho)\|x_N - x_* - \tfrac{1}{L}\nabla f(x_N)\|^2 \\
& + \tfrac{\rho L^2}{2(L-\mu)}\|x_N - x_* - \tfrac{1}{L}\nabla f(x_N)\|^2 + \tfrac{\rho L\mu}{2(L-\mu)}\|x_N - x_*\|^2 \\
& + \tfrac{\rho\mu}{L-\mu}\langle \nabla f(x_N), x_* - x_N\rangle + \tfrac{\rho}{2(L-\mu)}\|\nabla f(x_N)\|^2 \\
& - \tfrac{L^2}{2(L-\mu)}\|\rho(x_N - x_*) - \tfrac{\rho}{L}\nabla f(x_N)\|^2 \\
= & \Phi(x_{N+2}, x_{N+1}) - \rho\Phi(x_{N+1}, x_N) + \tfrac{L}{2}(1-\rho)\|x_{N+1} - x_N + \tfrac{1}{L}\nabla f(x_N)\|^2 \\
& + \tfrac{L}{2}\left(-(1-\rho) + \tfrac{\rho L(1-\rho)}{L-\mu}\right)\|x_N - x_* - \tfrac{1}{L}\nabla f(x_N)\|^2 \\
& + \tfrac{\rho L\mu}{2(L-\mu)}\|x_N - x_*\|^2 + \tfrac{\rho\mu}{L-\mu}\langle \nabla f(x_N), x_* - x_N\rangle + \tfrac{\rho}{2(L-\mu)}\|\nabla f(x_N)\|^2 \\
= & \Phi(x_{N+2}, x_{N+1}) - \rho\Phi(x_{N+1}, x_N) + \tfrac{L}{2}(1-\rho)\|x_{N+1} - x_N + \tfrac{1}{L}\nabla f(x_N)\|^2 \\
& + \tfrac{L}{2}\left(-(1-\rho) + \tfrac{\rho L(1-\rho)+\rho\mu}{L-\mu}\right)\|x_N - x_* - \tfrac{1}{L}\nabla f(x_N)\|^2 \\
& + \tfrac{\rho}{2L}\|\nabla f(x_N)\|^2 \\
= & \Phi(x_{N+2}, x_{N+1}) - \rho\Phi(x_{N+1}, x_N) + \tfrac{L}{2}(1-\rho)\|x_{N+1} - x_N + \tfrac{1}{L}\nabla f(x_N)\|^2 \\
& + \tfrac{\rho}{2L}\|\nabla f(x_N)\|^2,
\end{aligned}
\tag{3.39}
$$

where in the last equality we use that $-(1-\rho) + \frac{\rho L(1-\rho)+\rho\mu}{L-\mu} = 0$ when $\rho = \left(1 - \sqrt{\mu/L}\right)$. Finally, (3.39) implies that

$$0 \geq \Phi(x_{N+2}, x_{N+1}) - \rho\Phi(x_{N+1}, x_N),$$

which is the desired conclusion. ■

As previously noted, nesting inequalities from Proposition 3.2.6 together leads to the following convergence bound after $N$ iterations.

**Corollary 3.2.7.** *Let $d \in \mathbb{N}^*$, $f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$ with $0 < \mu < L < +\infty$, $N \in \mathbb{N}^*$ and $x_N$ the $N$-th iterate of Nesterov's method (3.34) initiated at $x_0 \in \mathbb{R}^d$. It holds that*

$$f(x_N) - f(x_*) \leq \left(1 - \sqrt{\tfrac{\mu}{L}}\right)^N \left[\tfrac{\mu L}{2(L-\mu)}\|x_0 - x_* - \tfrac{1}{L}\nabla f(x_0)\|^2 + f(x_0) - f(x_*)\right],$$

*where $x_* = \operatorname{argmin}_x f(x)$.*

*Proof.* Combining inequalities from Proposition 3.2.6 leads to

$$\Phi(x_{N+1}, x_N) \leq \left(1 - \sqrt{\tfrac{\mu}{L}}\right)\Phi(x_N, x_{N-1}) \leq \ldots \leq \left(1 - \sqrt{\tfrac{\mu}{L}}\right)^N \Phi(x_1, x_0).$$

We conclude by noticing that $\Phi(x_1, x_0) = \frac{\mu L}{2(L-\mu)}\|x_0 - x_* - \frac{1}{L}\nabla f(x_0)\|^2 + f(x_0) - f(x_*)$ and that $f(x_N) - f(x_*) \leq \Phi(x_{N+1}, x_N)$ . ■

**Remark 3.2.8.** *Note that more standard convergence results for Nesterov's method with constant momentum focus on bounding $f(y_N) - f(x_*)$. Indeed, in the case of composite minimization (i.e., the objective function writes as $f = h + g$ with $h \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$ and $g \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$) the iterates $\{x_k\}_k$ are not guaranteed to stay in $\mathbf{dom}\, f = \mathbf{dom}\, h$ which is not necessarily $\mathbb{R}^d$ in that case.*

**Remark 3.2.9.** *The result of Proposition 3.2.6 is not necessarily tight (i.e. there does not necessarily exist an entry such that there is equality). Indeed, we fixed the form of the potential functions and only search for parameters such that (3.29) is satisfied.*

We have seen on this example how we can design potentials to prove linear convergence guarantees. In the next example, we illustrate how to obtain potential functions to deal with sublinear rates.

**Example: Gradient descent** In Section 3.2.1 we used performance estimation to compute convergence guarantees after $N$ iterations of the gradient descent method with constant step size. Given $d \in \mathbb{N}^*$, $f \in \mathcal{F}_{0,\mathrm{L}}(\mathbb{R}^d)$ ($\mu = 0$), $\lambda \geq 0$ and $x_0 \in \mathbb{R}^d$, we consider updates

$$x_{k+1} = x_k - \lambda \nabla f(x_k), \tag{3.40}$$

which fits into (3.28) with $m = 1$. We look for potential functions of the form

$$\Phi_k(F, G) = F \underbrace{t_k(\mathbf{f}_k - \mathbf{f}_*)}_{a_k} + \mathrm{Tr}\left( \underbrace{[\mathbf{x}_k - \mathbf{x}_*, \mathbf{g}_k/L] S [\mathbf{x}_k - \mathbf{x}_*, \mathbf{g}_k/L]^T}_{A_k} G \right),$$

with $t_k \geq 0$ and $S \in \mathbf{S}_2^+$ such that (3.29) holds. With $S$ constant, the growth of the $\{t_k\}_k$ controls the convergence rate of $f(x_k) - f(x_*)$, therefore we are interested in finding the potentials with the largest $t_k$'s.

We search for the largest $t_{N+1}$ such that (3.29) holds given some fixed $t_N$, that is solving the following problem

$$\begin{aligned}
&\sup_{\substack{\nu \geq 0,\, S \succcurlyeq 0 \\ t_{N+1}}} \; t_{N+1} \\
&\quad \text{s.t. } \nu \geq 0,\; S \succcurlyeq 0, \\
&\qquad t_{N+1}(\mathbf{f}_{N+1} - \mathbf{f}_*) - t_N(\mathbf{f}_N - \mathbf{f}_*) - \tilde{F}(\nu) = 0, \\
&\qquad [\mathbf{x}_{N+1} - \mathbf{x}_*, \mathbf{g}_{N+1}/L]^T S [\mathbf{x}_{N+1} - \mathbf{x}_*, \mathbf{g}_{N+1}/L] \\
&\qquad - [\mathbf{x}_N - \mathbf{x}_*, \mathbf{g}_N/L]^T S [\mathbf{x}_N - \mathbf{x}_*, \mathbf{g}_N/L] - \tfrac{\tilde{G}(\nu) + \tilde{G}(\nu)^T}{2} \preccurlyeq 0,
\end{aligned} \tag{3.41}$$

with $\tilde{G}(\nu)$ and $\tilde{F}(\nu)$ defined as previously. It turns out that this problem is not well-posed (unbounded) as $S$ tends to grow to compensate with the maximization of $t_{N+1}$. Therefore, we introduce a normalization constraint on $S$ through the trace, as

$$\mathrm{Tr}(S) \leq 1,$$

preserving the linearity of the maximization problem (see §Codes for a matlab script solving this problem). From numerical trials we can find the following solutions

$$\begin{aligned}
&t_{N+1} = t_N + 2\lambda, \\
&\nu_{N,*} = 2\lambda, \nu_{N+1,N} = t_N + 2\lambda, \\
&S = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},
\end{aligned} \tag{3.42}$$

which leads to the same potential function as e.g Bansal and Gupta [2019, Theorem 3.3]. Finally, we obtain the next result using the proof mechanism presented in Section 3.2.2.

**Proposition 3.2.10.** *Let $d \in \mathbb{N}^*$, $0 < L < +\infty$, $f \in \mathcal{F}_{0,\mathrm{L}}(\mathbb{R}^d)$, $x_N \in \mathbb{R}^d$ be some iterate, $\lambda \in (0, \frac{1+\sqrt{5}}{2L}]$ be a step size and $t_N \geq 0$. It holds that*

$$\|x_{N+1} - x_*\|^2 + (t_N + 2\lambda)\left(f(x_{N+1}) - f(x_*)\right) \leq \|x_N - x_*\|^2 + t_N\left(f(x_N) - f(x_*)\right),$$

*where $x_* \in \operatorname{argmin}_x f(x)$ and $x_{N+1}$ is obtained using a gradient step with step size $\lambda$, that is*

$$x_{N+1} = x_N - \lambda \nabla f(x_N).$$

*Proof.* The proof consists in performing the following weighted sum of inequalities

- smoothness and convexity of $f$ between $x_N$ and $x_*$ with weight $\nu_{N,*} = 2\lambda$,
$$f(x_*) \geq f(x_N) + \langle \nabla f(x_N), x_* - x_N\rangle + \tfrac{1}{2L}\|\nabla f(x_{N+1})\|^2,$$
- smoothness and convexity of $f$ between $x_{N+1}$ and $x_N$ with weight $\nu_{N+1,N} = t_N + 2\lambda$,
$$f(x_N) \geq f(x_{N+1}) + \langle \nabla f(x_{N+1}), x_N - x_{N+1}\rangle + \tfrac{1}{2L}\|\nabla f(x_{N+1}) - \nabla f(x_N)\|^2.$$

This weighted sum takes the form

$$\begin{aligned} 2\lambda f(x_*) \geq & (t_N + 2\lambda) f(x_{N+1}) - t_N f(x_N) \\ & + 2\lambda\big[\langle \nabla f(x_N), x_* - x_N\rangle + \tfrac{1}{2L}\|\nabla f(x_N)\|^2\big] \\ & + (t_N + 2\lambda)\big[\langle \nabla f(x_{N+1}), x_N - x_{N+1}\rangle + \tfrac{1}{2L}\|\nabla f(x_{N+1}) - \nabla f(x_N)\|^2\big]. \end{aligned} \tag{3.43}$$

Introducing the quantities

$$\Phi_{N+1} = (t_N + 2\lambda)(f(x_{N+1}) - f(x_*)) + \|x_{N+1} - x_*\|^2,$$

and

$$\Phi_N = t_N(f(x_N) - f(x_*)) + \|x_N - x_*\|^2,$$

the weighted sum (3.43) becomes

$$\begin{aligned} 0 \geq & \Phi_{N+1} - \Phi_N - \|x_{N+1} - x_*\|^2 + \|x_N - x_*\|^2 \\ & + 2\lambda\big[\langle \nabla f(x_N), x_* - x_N\rangle + \tfrac{1}{2L}\|\nabla f(x_N)\|^2\big] \\ & + (t_N + 2\lambda)\big[\langle \nabla f(x_{N+1}), x_N - x_{N+1}\rangle + \tfrac{1}{2L}\|\nabla f(x_{N+1}) - \nabla f(x_N)\|^2\big]. \end{aligned} \tag{3.44}$$

We group together the terms in (3.44) that depend on $\nabla f(x_{N+1})$ as

$$\begin{aligned} & (t_N + 2\lambda)\big[\langle \nabla f(x_{N+1}), x_N - x_{N+1}\rangle + \tfrac{1}{2L}\|\nabla f(x_{N+1})\|^2 - \tfrac{1}{L}\langle \nabla f(x_{N+1}), \nabla f(x_N)\rangle\big] \\ = & \tfrac{t_N+2\lambda}{2L}\|\nabla f(x_{k+1}) + (L\lambda - 1)\nabla f(x_N)\|^2 - \tfrac{(t_N+2\lambda)(L\lambda-1)^2}{2L}\|\nabla f(x_N)\|^2, \end{aligned} \tag{3.45}$$

where we used the expression $x_{N+1} = x_N - \lambda \nabla f(x_N)$. We can then re-inject this equality in (3.44) and get

$$\begin{aligned} 0 \geq & \Phi_{N+1} - \Phi_N + \tfrac{t_N+2\lambda}{2L}\|\nabla f(x_{k+1}) + (L\lambda - 1)\nabla f(x_N)\|^2 - \tfrac{(t_N+2\lambda)(L\lambda-1)^2}{2L}\|\nabla f(x_N)\|^2 \\ & - \|x_{N+1} - x_*\|^2 + \|x_N - x_*\|^2 + 2\lambda\big[\langle \nabla f(x_N), x_* - x_N\rangle + \tfrac{1}{2L}\|\nabla f(x_N)\|^2\big] \\ & + (t_N + 2\lambda)\tfrac{1}{2L}\|\nabla f(x_N)\|^2 \\ = & \Phi_{N+1} - \Phi_N + \tfrac{t_N+2\lambda}{2L}\|\nabla f(x_{k+1}) + (L\lambda - 1)\nabla f(x_N)\|^2 - \tfrac{(t_N+2\lambda)(L\lambda-1)^2}{2L}\|\nabla f(x_N)\|^2 \\ & - \lambda^2\|\nabla f(x_N)\|^2 + \tfrac{2\lambda}{2L}\|\nabla f(x_N)\|^2 + (t_N + 2\lambda)\tfrac{1}{2L}\|\nabla f(x_N)\|^2 \\ = & \Phi_{N+1} - \Phi_N + \tfrac{t_N+2\lambda}{2L}\|\nabla f(x_{k+1}) + (L\lambda - 1)\nabla f(x_N)\|^2 \\ & + \tfrac{2\lambda+(t_N+2\lambda)-2L\lambda^2-(t_N+2\lambda)(L\lambda-1)^2}{2L}\|\nabla f(x_N)\|^2 \\ 0 \geq & \Phi_{N+1} - \Phi_N + \tfrac{2\lambda+(t_N+2\lambda)-2L\lambda^2-(t_N+2\lambda)(L\lambda-1)^2}{2L}\|\nabla f(x_N)\|^2, \end{aligned} \tag{3.46}$$

where we used the expression of $x_{N+1}$ in the first equality.

In order to obtain the desired result, that is

$$0 \geq \Phi_{N+1} - \Phi_N, \tag{3.47}$$

we need to show that $2\lambda + (t_N + 2\lambda) - 2L\lambda^2 - (t_N + 2\lambda)(L\lambda - 1)^2$ is nonnegative for the range of step sizes $\lambda$ considered. We can reformulate this quantify as follows

$$\begin{aligned} 2\lambda + (t_N + 2\lambda) - 2L\lambda^2 - (t_N + 2\lambda)(L\lambda - 1)^2 &= \lambda \left(2 - 2L\lambda + (t_N + 2\lambda)L(2 - L\lambda)\right) \\ &= \lambda \Big(2(1 + L\lambda - L^2\lambda^2) + t_N L(2 - L\lambda)\Big) \\ &\geq 0, \end{aligned}$$

as the polynomial $1 + X - X^2$ is nonnegative on $[0, \frac{1+\sqrt{5}}{2}]$ and $2 - L\lambda \geq 0$. This finally allows obtaining the conclusion (3.47). ■

As previously, this produces the following worst-case bound after $N$ iterations.

**Corollary 3.2.11.** *Let $d \in \mathbb{N}^*$, $f \in \mathcal{F}_{0,\mathrm{L}}(\mathbb{R}^d)$ with $0 < L < +\infty$, $N \in \mathbb{N}^*$ and $x_N$ the $N$-th iterate of Gradient descent (3.40) with step size $\lambda \in (0, \frac{1+\sqrt{5}}{2L}]$ initiated at $x_0 \in \mathbb{R}^d$. It holds that*

$$f(x_N) - f(x_*) \leq \tfrac{1}{2N\lambda}\|x_0 - x_*\|^2,$$

*where $x_* \in \operatorname{argmin}_x f(x)$.*

*Proof.* We can nest together inequalities from Proposition 3.2.10 with $t_N = 2\lambda N + t_0$ to get

$$t_N \left(f(x_N) - f(x_*)\right) + \|x_N - x_*\|^2 \leq \ldots \leq t_0 \left(f(x_0) - f(x_*)\right) + \|x_0 - x_*\|^2,$$

and the result follows from the choice $t_0 = 0$. ■

This convergence results after $N$ iterations appears slightly weaker that the conjecture and the numerical observations presented in the example of Figure 3.2. Indeed, Lyapunov theory often provides weaker guarantees as it focuses on the behavior of a small batch of iterates (i.e. does not take into account how previous iterates were computed) and relies on the (arbitrary) choice of the potential functions.

We have seen how Lyapunov arguments can be incorporated into the performance estimation framework, with the benefit of studying smaller semidefinite program ($m$ versus $N$) together with (often) simpler dual certificates. We refer the reader to Taylor and Bach [2019] for (comprehensive) developments on the use of potential functions in PEPs.

**Design of optimization methods** We end this introduction on the performance estimation framework by briefly mentioning the possibility of designing optimization methods with optimized worst-case guarantees using PEPs.

In the previous part, we have seen that we could introduce some degrees of freedom in potential functions. Similarly, we can add degrees of freedom in the methods coefficients and search for methods parameters that make the worst-case bounds the smallest possible. However, this may produce nonlinearity in the semidefinite reformulations. The main problem of methods design in this context, consists in removing this nonlinearity by slightly modifying the problem to solve. Several technical "tricks" has been developed using e.g. change of variables, Schur complements, and/or relaxations (see e.g. Drori and Teboulle [2014], Kim and Fessler [2016], Taylor and Drori [2021] or Chapter 5).

# Chapter 4

# Worst-Case Analyses of Adaptive Methods: Study of Polyak Step Sizes

In this chapter, we focus on unconstrained optimization problems of the form

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

where $f$ is strongly convex and has a Lipschitz continuous gradient with respect to the Euclidean norm. Very broadly speaking, the current numerical toolbox to solve these convex minimization problems contains two types of methods. On one hand, simple numerical schemes with explicit albeit conservative theoretical guarantees. These include gradient methods and their accelerated variants, and require knowing problem *parameters*, such as strong convexity parameters, or Hölderian error bounds [Bolte et al., 2007]. On the other hand, *adaptive methods*, such as conjugate gradients or quasi-Newton, adapting much better to the objective function by estimating some of its regularity properties. For these methods, we typically have few justifications for their improved performances. In the following, we study worst-case guarantees of some adaptive first-order methods, and in particular some methods based on "Polyak steps"[Polyak, 1987].

**Contributions** Our contributions can be summarized as follows:

(i) We provide new worst-case bounds for variants of gradient descent with Polyak steps.

(ii) We develop and analyze an accelerated gradient method with Polyak steps based momentum.

(iii) We show how Performance Estimation Problems (PEPs) [Drori and Teboulle, 2014, Taylor et al., 2017c] can be used to analyze some adaptive algorithms.

**Organization** We start this chapter by an introduction and a short review of adaptive first-order methods in Section 4.1. Then, in Section 4.2 we study worst-case behaviors of variants of gradient descent with Polayk step sizes, and accelerated versions in Section 4.3. In Section 4.4, we detail the analysis mechanisms behind the bounds provided in previous sections, and present numerical analyses of several adaptive methods in Section 4.5. Finally, we present in Section 4.6 numerical experiments on the accelerated gradient method with Polyak steps based momentum, and conclude in Section 4.7.

## 4.1 Introduction

Empirically, adaptive optimization methods often perform significantly better than their parametric counterparts, and, by nature, require much less tuning. For example, roughly estimating regularity constants on-the-fly and plugging these estimates in parametric algorithms often produces fast algorithms with no theoretical guarantees. This phenomenon is illustrated in Figure 4.1 on logistic regression.

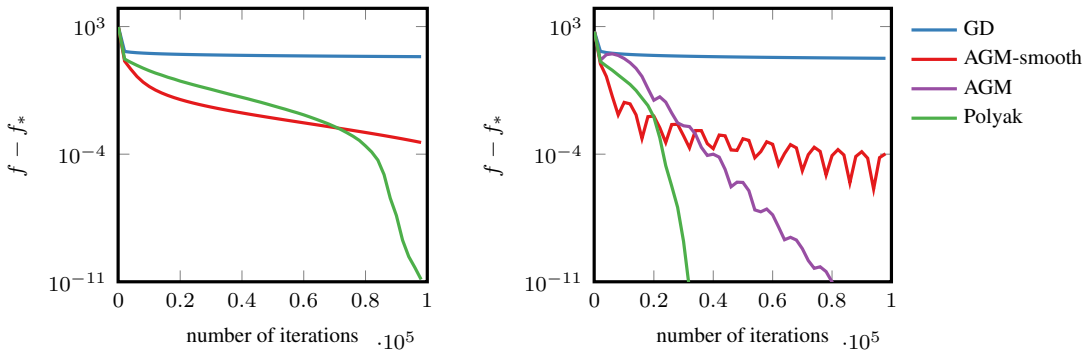


Figure 4.1: Convergence of gradient descent (GD), accelerated gradient method for smooth optimization (AGM-smooth) [Nesterov, 1983], accelerated gradient method with constant momentum (AGM)—described below as Algorithm 4.3 with (Const-mom)—where the momentum is set using the value of the regularization parameter and gradient method with Polyak steps (Polyak). Experiments on regularized logistic regression for the Sonar dataset without any tuning of the methods. Left: regularization parameter $10^{-7}$. Right: regularization parameter $10^{-4}$. For Polyak steps, the best iterate is displayed. Observe that Polyak method is a (non-accelerated) adaptive method, which performs comparatively well against accelerated schemes.

Although many advances have been made in designing optimization schemes adaptive to some types of parameters (e.g., Lipschitz constants, see discussions below), these results still leave a huge gap between theory and practice (as in Figure 4.1). In particular, estimating strong convexity coefficients while preserving convergence guarantees remains a challenging issue. Restart schemes are probably the most effective option among existing approaches for adapting to this type of parameters and do provide improved complexity estimates without any knowledge of strong convexity parameters, at the expense of a log scale grid search. However, while on paper the complexity of these schemes is nearly optimal, the presence of an outer loop clearly limits their practical effectiveness and their capacity to adapt to the function's local regularity, which leaves a lot of margin for improvement, on the numerical front. Producing single loop algorithms adapting to local strong convexity (or Hölderian error bounds) and have nearly optimal complexity bounds is an important open problem which is the main focus of this work.

Here, we study the complexity of adaptive methods using Polyak steps, estimating the momentum term using information on the optimum objective value $f_*$ instead of the strong convexity constant. In some scenarios, such as "interpolation" in machine learning problems, the value of $f_*$ is known a priori (usually zero), and estimating it is much easier than estimating strong convexity, see e.g., Asi and Duchi [2019] for a recent discussion on these model assumptions.

The obvious next research question in this direction is to substitute knowledge on $f_*$ by weaker bounds. A first step in this direction is for example Hazan and Kakade [2019] which uses successive refinements of a lower bound on $f_*$. However, it seems that in the case of linear convergence, the extra

cost of this procedure might be prohibitive.

## Related works

**Gradient and accelerated gradient methods.** For smooth optimization problems, simple line search strategies provide accelerated algorithms that adapt to the local gradient Lipschitz constant [Nesterov, 2013] and explicit adaptive complexity bounds can be derived for certain variants using the mean root Lipschitz constant [Scheinberg et al., 2014].

**Restarts.** For smooth and strongly convex optimization problems (or more generally problems satisfying Hölderian error bounds), accelerated methods with optimal complexity bounds require knowledge of the strong convexity constant to compute iterates [Nesterov, 2013, 2018]. In particular, Arjevani and Shamir [2016] show that this information is necessary when using oblivious steps. This quantity can be hard to estimate and a lot of effort has been put in the development of adaptive optimization methods preserving fast convergence rates [Lin and Xiao, 2014, Fercoq and Qu, 2016, Roulet and d'Aspremont, 2020]. All these works are based on restart strategies [O'Donoghue and Candes, 2015, Nesterov, 2013] and although they exhibit fast theoretical convergence rates, they often contain parameters that have to be tuned in order to get good practical results, or require additional information on the function itself (e.g., its minimum $f_*$). Once again, while on paper the complexity of restart schemes is nearly optimal, the presence of an outer loop generally limits their capacity to adapt to the function's local regularity and significantly affects empirical performance.

**Quasi-Newton methods.** An important family of adaptive algorithms is composed of quasi-Newton methods. As the name suggests, these methods try to mimic the behavior of Newton schemes, by constructing an estimate of the hessian at the current point, using previous gradients. The most notable quasi-Newton method is certainly L-BFGS [Liu and Nocedal, 1989]. These commonly used algorithms exhibit very fast empirical converge rates but only local improvements over gradient descent have been proven at this point.

**Conjugate gradient methods.** Conjugate gradient methods are probably among the most famous examples of adaptive algorithm. Firstly introduced for quadratic minimization [Hestenes and Stiefel, 1952], and motivated by nice theoretical guarantees (such as finite-time convergence), many variants have been introduced for going beyond quadratics [Fletcher and Reeves, 1964, Polyak, 1969, Fletcher, 1987]—see, for example, the nice survey Hager and Zhang [2006]. Roughly speaking, at each iteration, the method constructs an update direction based on the gradient at the current iterate, and on the knowledge of the previous search directions. The next iterate is obtained by line-search in the update direction. Whereas conjugate gradient methods are widely used in practice (e.g Rodi and Mackie [2001], Volkwein [2004], Zhao et al. [2015]), and perform very well when they applies, there are barely any non-asymptotic convergence guarantees for those methods beyond unconstrained quadratic minimization.

**Polyak step sizes.** When the optimal value of the objective function value is known, a well-known adaptive strategy consists in using the so-called "Polyak step sizes"—see e.g., Polyak [1987, Section 5.3.2] or Nedic and Bertsekas [2001], Boyd et al. [2003]. The method consists in iterating gradient steps with step sizes proportional to the primal gap at the current iterate. As opposed to most adaptive

gradient methods mentioned above, this method comes with explicit theoretical properties, even beyond the quadratic optimization case.

**Barzilai-Borwein step sizes.** The Barzilai-Borwein [Barzilai and Borwein, 1988, Fletcher, 2005] method consists in gradient steps with adaptive step sizes. It is another case with complete theory for quadratic optimization, but barely any performance guarantees in non-quadratic cases (it is even known to diverge on some problem instances).

**Adaptive gradient steps** In Malitsky and Mishchenko [2019] the authors developed a step size policy that adapts to the local geometry, together with nice theoretical guarantees.

### Preliminaries

We denote $f_*$ the minimum of $f$. Let $0 \leq \mu < L < +\infty$, the class of $L$-smooth and $\mu$-strongly convex functions is denoted $\mathcal{F}_{\mu,L}(\mathbb{R}^d)$. Functions in this class satisfy (see e.g., Nesterov [2018]) $\forall x, y \in \mathbb{R}^d$:

$$\begin{aligned} f(y) &\leq f(x) + \langle \nabla f(x), y - x \rangle + \tfrac{L}{2}\|y - x\|^2 \quad \text{(smoothness)}, \\ f(y) &\geq f(x) + \langle \nabla f(x), y - x \rangle + \tfrac{\mu}{2}\|y - x\|^2 \quad \text{(strong convexity)}. \end{aligned}$$

We recall the following inequality satisfied by smooth and strongly convex functions.

**Lemma 4.1.1.** *[Taylor et al., 2017c, Theorem 4] Given $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, for any $(x, y) \in \mathbb{R}^d \times \mathbb{R}^d$*

$$\begin{aligned} &f(x) - f(y) + \nabla f(x)^T (y - x) + \tfrac{1}{2L}\|\nabla f(x) - \nabla f(y)\|^2 \\ &+ \tfrac{\mu}{2(1-\frac{\mu}{L})}\|x - y - \tfrac{1}{L}(\nabla f(x) - \nabla f(y))\|^2 \leq 0 \end{aligned}$$

The proof presented in this chapter mostly rely on these inequalities (or weaker versions of it).

## 4.2 Classical Polyak steps and variants

Let us start with complexity bounds for gradient methods with Polyak steps for smooth and strongly convex optimization problems. Note that Polyak step sizes are usually discussed in the nondifferentiable setting—see Polyak [1987, Section 5.3.2] or Nedic and Bertsekas [2001], Boyd et al. [2003]. We first recall the complexity of the gradient method with Polyak steps in the smooth strongly convex case, then derive similar bounds for two variants. For the first variant, we scale the steps by a factor two compared to standard Polyak steps, yielding a simple convergence proof with slightly improved theoretical guarantees. The second variant is a descent method, where the complexity bound is written in terms of the primal gap. We delay a full discussion of the proof mechanisms to Section 4.4.

**Adaptive gradient method** (Algorithm 4.2)

**Input:**

- Objective function: $f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$.
- Initial guess: $x_0 \in \mathbb{R}^d$.

- Optimal objective value: $f_* \in \mathbb{R}$ .

**Run:**

For $k = 0, \dots$ :

Compute $\gamma_k$

$x_{k+1} = x_k - \gamma_k \nabla f(x_k)$

End For

**Output:** $x_{k+1}$

| | | |
|---|---|---|
| Regular Polyak steps: | $\gamma_k = \frac{f(x_k)-f_*}{\|\nabla f(x_k)\|^2}$ | (Polyak) |
| Polyak steps, variant I: | $\gamma_k = 2\frac{f(x_k)-f_*}{\|\nabla f(x_k)\|^2}$ | (Variant I) |
| Polyak steps, variant II: | $\gamma_k = \left(2 - \frac{\|\nabla f(x_k)\|^2}{2L(f(x_k)-f_*)}\right)/L$ | (Variant II) |

The classical step size rule (Polyak) was mostly studied in the nonsmooth convex case [Polyak, 1987]. For smooth strongly convex problems, it is known (see e.g., Hazan and Kakade [2019]) that

$$f(x_N) - f_* \leq (1 - \tfrac{\mu}{L})^N \tfrac{L\|x_0 - x_*\|^2}{2}. \tag{4.1}$$

The two following propositions show that different step sizes policies (namely (Variant I) and (Variant II)) produce slightly improved convergence rates, matching the best known rates for gradient methods with known $\mu$ and $L$. The $\gamma_k$ are always well defined except when $x_k$ has a zero gradient, in this case we can simply stop the method since we have reached optimality. When it is well defined, $\gamma_k \in [\frac{1}{L}, \frac{1}{\mu}]$ for (Variant I) and $\gamma_k \in [\frac{1}{L}, \frac{2-\mu/L}{L}]$ for (Variant II).

### 4.2.1 Study of adaptive gradient method with Variant I

First, let us state that if we seek to decrease the distance to the optimal point, (Variant I) provides a rate that matches that of gradient descent with optimal (non-adaptive) step sizes [Nesterov, 2018] in the worst-case.

**Proposition 4.2.1** (Variant I)**.** *Let $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ and consider Algorithm 4.2 with step sizes* (Variant I)*. Then, for any $x_0 \in \mathbb{R}^d$ and $N \in \mathbb{N}$, such that the sequence $\{\gamma_k\}_k$ is well defined, it holds that*

$$\|x_N - x_*\|^2 \leq \left(\prod_{k=0}^{N-1} \rho(\gamma_k)\right) \|x_0 - x_*\|^2,$$

*where $x_* \in \operatorname{argmin}_x f(x)$, $\rho(\gamma) = \frac{(\gamma L - 1)(1 - \gamma\mu)}{\gamma(L+\mu) - 1}$, and $\max_{\gamma \in [\frac{1}{L}, \frac{1}{\mu}]} \rho(\gamma) = \frac{(L-\mu)^2}{(L+\mu)^2}$. Otherwise $\nabla f(x_k) = 0$ with $k \in [0, N]$.*

*In addition, for all $d \geq 2$, $x_0 \in \mathbb{R}^d$ and $0 < \mu < L$, there exists $f_{x_0} \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ such that $x_N$ the output of Algorithm 4.2 with step sizes (Variant I) applied to $f_{x_0}$ satisfies*

$$\|x_N - x_*\|^2 = \left(\tfrac{L-\mu}{L+\mu}\right)^{2N} \|x_0 - x_*\|^2.$$

For readability purposes the proof of Proposition 4.2.1 is cut in half.

**Convergence bound** First, we present the proof of the convergence bound.

*Proof.* For proving the desired result, it is only necessary to consider a single iteration of Algorithm 4.2 with (Variant I). We use the following (in)equalities obtained from Lemma 4.1.1:

- smoothness and strong convexity between $x_k$ and $x_*$, with multiplier $\lambda_1 = \frac{2\gamma_k(\gamma_k L-1)}{\gamma_k(L+\mu)-1}$:

$$f(x_k) - f_* + \nabla f(x_k)^T(x_* - x_k) + \tfrac{1}{2L}\|\nabla f(x_k)\|^2 + \tfrac{\mu}{2(1-\frac{\mu}{L})}\|x_k - x_* - \tfrac{1}{L}\nabla f(x_k)\|^2 \leq 0,$$

- smoothness and strong convexity between $x_*$ and $x_k$, with multiplier $\lambda_2 = \frac{2\gamma_k(1-\gamma_k\mu)}{\gamma_k(L+\mu)-1}$:

$$f_* - f(x_k) + \tfrac{1}{2L}\|\nabla f(x_k)\|^2 + \tfrac{\mu}{2(1-\frac{\mu}{L})}\|x_k - x_* - \tfrac{1}{L}\nabla f(x_k)\|^2 \leq 0,$$

- definition of the step size policy, with multiplier $\lambda_3 = \frac{\gamma_k(2-\gamma_k(L+\mu))}{\gamma_k(L+\mu)-1}$:

$$2(f(x_k) - f_*) - \gamma_k\|\nabla f(x_k)\|^2 = 0.$$

Given that $\lambda_1, \lambda_2 \geq 0$ (since $\frac{1}{L} \leq \gamma_k \leq \frac{1}{\mu}$), the following weighted sum is a valid inequality:

$$\begin{aligned} 0 \geq &\lambda_1 \left[f(x_k) - f_* + \nabla f(x_k)^T(x_* - x_k) + \tfrac{1}{2L}\|\nabla f(x_k)\|^2 + \tfrac{\mu}{2(1-\frac{\mu}{L})}\|x_k - x_* - \tfrac{1}{L}\nabla f(x_k)\|^2\right] \\ &+ \lambda_2 \left[f_* - f(x_k) + \tfrac{1}{2L}\|\nabla f(x_k)\|^2 + \tfrac{\mu}{2(1-\frac{\mu}{L})}\|x_k - x_* - \tfrac{1}{L}\nabla f(x_k)\|^2\right] \\ &+ \lambda_3 \left[2(f(x_k) - f_*) - \gamma_k\|\nabla f(x_k)\|^2\right]. \end{aligned}$$

Using the fact that $x_{k+1} = x_k - \gamma_k\nabla f(x_k)$, this weighted sum can be reformulated exactly as

$$\|x_{k+1} - x_*\|^2 - \rho(\gamma_k)\|x_k - x_*\|^2 \leq 0$$

(one can verify that both expressions are equal) with $\rho(\gamma) = \frac{(\gamma L-1)(1-\gamma\mu)}{\gamma(L+\mu)-1}$. Therefore, after $N$ iterations, we get

$$\|x_N - x_*\|^2 \leq \left(\prod_{i=0}^{N-1}\rho(\gamma_i)\right)\|x_0 - x_*\|^2.$$

In addition, distance to optimality decreases, in the worst-case, with rate $\max_\gamma \rho(\gamma)$, with

$$\tfrac{(L-\mu)^2}{(L+\mu)^2} = \max\left\{\rho(\gamma) \mid \tfrac{1}{L} \leq \gamma \leq \tfrac{1}{\mu}\right\}.$$

because $\rho(\gamma)$ is a concave function of $\gamma$ on the interval $[\frac{1}{L}, \frac{1}{\mu}]$, as $\rho''(\gamma) = -\frac{2L\mu}{(\gamma(L+\mu)-1)^3} \leq 0$, whose maximum is attained at $\gamma_* = \frac{2}{L+\mu}$. Note that substituting the expression of $\gamma_k$ inside the interpolation inequalities, instead of using it as an independent equality constraints, yields a considerably less tractable result. ■

**Tightness** Now, we focus on the tightness result in Proposition 4.2.1. We construct a 2-dimensional quadratic function on which Algorithm 4.2 with step sizes (Variant I) reaches the maximal convergence bound.

*Proof.* First, we consider the case $d = 2$ and $x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$. We define the quadratic function $f$ as

$$f(x) := \tfrac{1}{2} x^T H x,$$

with

$$H = \frac{1}{L+\mu} \begin{pmatrix} 2L\mu & -(L-\mu)\sqrt{L\mu} \\ -(L-\mu)\sqrt{L\mu} & L^2+\mu^2 \end{pmatrix}.$$

We can check that $f$ belongs to $\mathcal{F}_{\mu,L}(\mathbb{R}^2)$. Indeed, $\mathrm{Tr}(H) = L + \mu$ and $\det(H) = L\mu$ which implies that $H$ admits $L$ and $\mu$ as eigenvalues. Moreover, $f$ reaches its minimal value $f_* = 0$ in $x_* = 0$.

Then, we show that for $\{x_k\}_k$ the sequence of iterates defined by

$$x_{k+1} = x_k - 2\tfrac{f(x_k)-f_*}{\|\nabla f(x_k)\|^2} \nabla f(x_k), \quad \text{for } k \geq 0,$$

satisfies

$$\begin{aligned} x_{2k+1} &= \tfrac{(L-\mu)^2}{(L+\mu)^2}^{2k} x_1 \\ x_{2k+2} &= \tfrac{(L-\mu)^2}{(L+\mu)^2}^{2k+2} x_0. \end{aligned} \tag{4.2}$$

with $\|x_1 - x_*\|^2 = \frac{(L-\mu)^2}{(L+\mu)^2} \|x_0 - x_*\|^2$, which implies the awaited result

$$\|x_k - x_*\|^2 = \left(\tfrac{(L-\mu)^2}{(L+\mu)^2}\right)^N \|x_0 - x_*\|^2, \quad \text{for } k \geq 0. \tag{4.3}$$

We start by studying the form of the first Polyak step size, that is

$$\begin{aligned} 2\tfrac{f(x_0)-f_*}{\|\nabla f(x_0)\|^2} &= 2\tfrac{L\mu}{L+\mu} \tfrac{(L+\mu)^2}{4L^2\mu^2+L\mu(L-\mu)^2} \\ &= 2\tfrac{(L+\mu)}{4L\mu+(L-\mu)^2} \\ &= \tfrac{2}{L+\mu}. \end{aligned}$$

Therefore, we can express $x_1$ as

$$x_1 = \begin{pmatrix} 1 - \frac{2}{L+\mu}\frac{2L\mu}{L+\mu} \\ \frac{2}{L+\mu}\frac{(L-\mu)\sqrt{L\mu}}{L+\mu} \end{pmatrix} = \tfrac{(L-\mu)}{(L+\mu)^2} \begin{pmatrix} L-\mu \\ 2\sqrt{L\mu} \end{pmatrix}. \tag{4.4}$$

And finally,

$$\begin{aligned} \|x_1 - x_*\|^2 &= \tfrac{(L-\mu)^4+4L\mu(L-\mu)^2}{(L+\mu)^4} \\ &= \tfrac{(L-\mu)^2}{(L+\mu)^2} \tfrac{(L-\mu)^2+4L\mu}{(L+\mu)^4} = \tfrac{(L-\mu)^2}{(L+\mu)^2}. \end{aligned}$$

Then, we proceed similarly and observe (after some simplifications) that the second Polyak step size is

$$2\tfrac{f(x_1)-f_*}{\|\nabla f(x_1)\|^2} = \tfrac{2}{L+\mu}. \tag{4.5}$$

Based on it, we can express $x_2$ as

$$x_2 = \begin{pmatrix} \frac{(L-\mu)^2}{(L+\mu)^2} - \frac{2}{L+\mu}0 \\ 2\frac{(L-\mu)\sqrt{L\mu}}{(L+\mu)^2} - \frac{2}{L+\mu}\frac{(L-\mu)\sqrt{L\mu}}{L+\mu} \end{pmatrix} = \begin{pmatrix} \frac{(L-\mu)^2}{(L+\mu)^2} \\ 0 \end{pmatrix} = \frac{(L-\mu)^2}{(L+\mu)^2} x_0.$$

Using the quadratic form of $f$ (i.e., $f(\alpha x) = \alpha^2 f(x)$ and $\nabla f(\alpha x) = \alpha \nabla f(x)$ for all $\alpha \in \mathbb{R}$ and $x \in \mathbb{R}^2$), Polyak step sizes remain equal to $\frac{2}{L+\mu}$ throughout iterations and equations (4.2), (4.3) hold.

In order to get the result for any $x_0 \neq 0$, we can consider $O$ the rotation matrix such that $Ox_0 = \|x_0\|e_1$ with $e_1$ the vector of $\mathbb{R}^2$ with 1 is first position and 0 in second position. Applying Algorithm (4.2) with step sizes (Variant I) to the function $g(x) = f(Ox)$ produces iterates that also satisfy (4.3).

Finally, in higher dimensions, we can simply consider a quadratic function $f$ such that

$$f(x) = \frac{1}{2} x^T \begin{pmatrix} H & 0 & \dots & 0 \\ 0 & \mu & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & \dots & 0 & \mu \end{pmatrix} x,$$

and for $x_0 \in \mathbb{R}^d$, define similarly $g(x) = f(Ox)$ with $O$ the rotation matrix such that $Ox_0 = e_1$ the basis vector of $\mathbb{R}^d$ with 1 in first position and 0 elsewhere. ■

### 4.2.2 Study of adaptive gradient method with Variant II

If on the other hand we seek to decrease the primal gap, (Variant II) provides a rate that matches that of gradient descent with exact line search [Klerk et al., 2017], at the expense of knowledge on L.

**Proposition 4.2.2** (Variant II)**.** *Let $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, consider Algorithm 4.2 with step sizes* (Variant II)*. Then, for any $x_0 \in \mathbb{R}^d$ and $N \in \mathbb{N}$, such that the sequence $\{\gamma_k\}_k$ is well defined, it holds that*

$$f(x_N) - f_* \leq \left( \prod_{k=0}^{N-1} \rho(\gamma_k) \right) (f(x_0) - f_*),$$

*where $f_* = \min_x f(x)$, $\rho(\gamma) = (L\gamma - 1)\left(L\gamma(3 - \gamma(L+\mu)) - 1\right)$, and $\max_{\gamma \in [\frac{1}{L}, \frac{2L-\mu}{L^2}]} \rho(\gamma) = \frac{(L-\mu)^2}{(L+\mu)^2}$. Otherwise $\nabla f(x_k) = 0$ with $k \in [0, N]$.*

*Proof.* Let us consider a single iteration of Algorithm 4.2, with step sizes (Variant II). The proof is a consequence of the following combination of inequalities obtained from Lemma 4.1.1:

- smoothness and strong convexity between $x_k$ and $x_*$, with multiplier $\lambda_1 = \gamma_k \mu(L\gamma_k - 1)$:

$$f(x_k) - f_* + \nabla f(x_k)^T(x_* - x_k) + \frac{1}{2L}\|\nabla f(x_k)\|^2 + \frac{\mu}{2(1-\frac{\mu}{L})}\|x_k - x_* - \frac{1}{L}\nabla f(x_k)\|^2 \leq 0,$$

- smoothness and strong convexity between $x_{k+1}$ and $x_*$, with multiplier $\lambda_2 = \gamma_k \mu$:

$$f(x_{k+1}) - f_* + \nabla f(x_{k+1})^T(x_* - x_{k+1}) + \frac{1}{2L}\|\nabla f(x_{k+1})\|^2 + \frac{\mu}{2(1-\frac{\mu}{L})}\|x_{k+1} - x_* - \frac{1}{L}\nabla f(x_{k+1})\|^2 \leq 0,$$

- smoothness and strong convexity between $x_{k+1}$ and $x_k$, with multiplier $\lambda_3 = 1 - \gamma_k \mu$:

$$\begin{aligned} f(x_{k+1}) - f(x_k) + \nabla f(x_{k+1})^T (x_k - x_{k+1}) + \tfrac{1}{2L} \|\nabla f(x_{k+1}) - \nabla f(x_k)\|^2 \\ + \tfrac{\mu}{2(1-\frac{\mu}{L})} \|x_{k+1} - x_k - \tfrac{1}{L}(\nabla f(x_{k+1}) - \nabla f(x_k))\|^2 \leq 0, \end{aligned}$$

- definition of the step size policy, with multiplier $\lambda_4 = \frac{\gamma_k}{2}((L+\mu)\gamma_k - 2)$:

$$(2L^2\gamma_k - 4L)(f(x_k) - f_*) + \|\nabla f(x_k)\|^2 = 0.$$

Given that $\lambda_1, \lambda_2, \lambda_3 \geq 0$ (due to $\frac{1}{L} \leq \gamma_k \leq \frac{2-\frac{\mu}{L}}{L}$), the following weighted sum is a valid inequality:

$$\begin{aligned} 0 \geq & \lambda_1 \left[ f(x_k) - f_* + \nabla f(x_k)^T (x_* - x_k) + \tfrac{1}{2L} \|\nabla f(x_k)\|^2 + \tfrac{\mu}{2(1-\frac{\mu}{L})} \|x_k - x_* - \tfrac{1}{L} \nabla f(x_k)\|^2 \right] \\ & + \lambda_2 \Big[ f(x_{k+1}) - f_* + \nabla f(x_{k+1})^T (x_* - x_{k+1}) + \tfrac{1}{2L} \|\nabla f(x_{k+1})\|^2 \\ & \quad + \tfrac{\mu}{2(1-\frac{\mu}{L})} \|x_{k+1} - x_* - \tfrac{1}{L} \nabla f(x_{k+1})\|^2 \Big] \\ & + \lambda_3 \Big[ f(x_{k+1}) - f(x_k) + \nabla f(x_{k+1})^T (x_k - x_{k+1}) + \tfrac{1}{2L} \|\nabla f(x_{k+1}) - \nabla f(x_k)\|^2 \\ & \quad + \tfrac{\mu}{2(1-\frac{\mu}{L})} \|x_{k+1} - x_k - \tfrac{1}{L}(\nabla f(x_{k+1}) - \nabla f(x_k))\|^2 \Big] \\ & + \lambda_4 \left[ (2L^2\gamma_k - 4L)(f(x_k) - f_*) + \|\nabla f(x_k)\|^2 \right]. \end{aligned}$$

Using the expression $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$ (without substituting the expression of $\gamma_k$, whose value is encoded through the last equality of the list), this weighted sum can be rewritten exactly as

$$\begin{aligned} 0 \geq & f(x_{k+1}) - f_* - \rho(\gamma_k)(f(x_k) - f_*) \\ & + \tfrac{1}{2(L-\mu)} \|\nabla f(x_{k+1}) - L\mu\gamma_k(x_k - x_*) + (\gamma_k(L+\mu) - 1)\nabla f(x_k)\|^2 \end{aligned}$$

with $\rho(\gamma) = (L\gamma - 1)\left(L\gamma(3 - \gamma(L+\mu)) - 1\right)$ which, in turns, give

$$\begin{aligned} f(x_{k+1}) - f_* \leq & \rho(\gamma_k)(f(x_k) - f_*) \\ & - \tfrac{1}{2(L-\mu)} \|\nabla f(x_{k+1}) - L\mu\gamma_k(x_k - x_*) + (\gamma_k(L+\mu) - 1)\nabla f(x_k)\|^2 \\ \leq & \rho(\gamma_k)(f(x_k) - f_*). \end{aligned}$$

Therefore, after $N$ iterations, we get

$$f(x_N) - f_* \leq \left( \prod_{i=0}^{N-1} \rho(\gamma_i) \right) (f(x_0) - f_*).$$

Finally, the worst-case convergence rate is $\max_\gamma \rho(\gamma)$ on the interval $[\frac{1}{L}, \frac{2-\mu/L}{L}]$, for which

$$\tfrac{(L-\mu)^2}{(L+\mu)^2} = \max \left\{ \rho(\gamma) \mid \tfrac{1}{L} \leq \gamma \leq \tfrac{2-\mu/L}{L} \right\}.$$

The proof follows from the following steps:

- First, on the boundaries of the interval: (i) $\rho(\frac{1}{L}) = 0$ and (ii) $\rho(\frac{2-\frac{\mu}{L}}{L}) = \frac{(L-\mu)^4}{L^4} \leq \frac{(L-\mu)^2}{(L+\mu)^2}$.
- Secondly, in the interior of the interval: $\rho'(\gamma) = L(3L\gamma - 2)(2 - (L+\mu)\gamma)$ is zero at $\gamma_* = \frac{2}{L+\mu}$ (inside the interval).
- Therefore $\rho(\gamma_*) = \frac{(L-\mu)^2}{(L+\mu)^2}$ and this is the maximum on the interval.

■

In the following section, we study variants of those methods, where we aim to speed up convergence by incorporating a momentum term. Those methods follow in spirit the line of works on Nesterov's acceleration [Nesterov, 2013], where we supersede knowledge of $\mu$ by that of $f_*$.

## 4.3 Acceleration with Polyak momentum

In the following, AGM refers to the Accelerated Gradient Method with momentum introduced by Nesterov [Nesterov, 1983, 2018]. We are interested in optimizing a function $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ without any information on the strong convexity constant $\mu$. However, as in the Polyak gradient method, we rely on the knowledge of $f_*$. We describe a single loop adaptive accelerated method (i.e. without restarts), with convergence rate of order $1 - (\mu/L)^{3/4}$, compared with $1 - \mu/L$ for gradient descent, and $1 - (\mu/L)^{1/2}$ for its accelerated version with perfect knowledge of $\mu$.

**Accelerated gradient method (AGM)** (Algorithm 4.3)

**Input:**

- Objective function: $f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$.
- Initial guess: $x_0 \in \mathbb{R}^d$.
- Optimal objective value: $f_* \in \mathbb{R}$.
- Smoothness parameter: $L \in \mathbb{R}^*$.

**Initialization:**

$y_0 = x_0$

**Run:**

For $k = 0, \ldots$ :

$y_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$

Compute $\tilde{\mu}_k$ and $\beta_k = \frac{\sqrt{L}-\sqrt{\tilde{\mu}_k}}{\sqrt{L}+\sqrt{\tilde{\mu}_k}}$

$x_{k+1} = y_{k+1} + \beta_k(y_{k+1} - y_k)$

End For

**Output:** $y_{k+1}$

$$\text{Constant momentum:} \quad \tilde{\mu}_k = \mu \tag{Const-mom}$$

$$\text{Polyak Acc., variant I:} \quad \tilde{\mu}_k = \frac{\|\nabla f(y_{k+1})\|^2}{2(f(y_{k+1})-f_*)}, \tag{Acc. Variant I}$$

$$\text{Polyak Acc., variant II:} \quad \tilde{\mu}_k = \begin{cases} +\infty & \text{if } k = -1 \\ \min\left(\tilde{\mu}_{k-1}, \frac{\|\nabla f(y_{k+1})\|^2}{2(f(y_{k+1})-f_*)}\right) & \text{otherwise} \end{cases} \tag{Acc. Variant II}$$

Algorithm 4.3 is based on the AGM algorithm [Nesterov, 2018], in which the knowledge of $\mu$ is essential to set the constant momentum term $\beta_k = \beta_* = (\sqrt{L} - \sqrt{\mu})/(\sqrt{L} + \sqrt{\mu})$. Common convergence guarantees require a lower bound on the strong convexity.

### 4.3.1 Robustness of AGM

As a first step towards producing adaptive versions of AGM, Lemma 4.3.1 and Corollary 4.3.2 below guarantee that AGM with any momentum factor $\beta_k$ in $[0, 1]$ converges at least as fast as the classical gradient method.

**Lemma 4.3.1** (Convergence of AGM with bad momentum)**.** *Let $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, some iteration number $k \in \mathbb{N}$, and consider Algorithm 4.3 with $\beta_k \in [0, 1]$. Then, for any $x_k, y_k \in \mathbb{R}^d$, it holds that*

$$\Phi(x_{k+1}, y_{k+1}) \leq \rho\Phi(x_k, y_k) \tag{4.6}$$

*where $\Phi(x, y) = \frac{L-\mu}{2}\|x - y\|^2 + f(y) - f_*$, $f_* = \min_x f(x)$ and $\rho = 1 - \frac{\mu}{L}$.*

*Proof.* We use the notation $\rho = 1 - \mu/L$ in this proof. It consists in combining the following inequalities obtained from Lemma 4.1.1:

- smoothness and strong convexity between $x_k$ and $y_k$ with multiplier $\lambda_1 = \rho$:

$$\begin{aligned} f(x_k) - f(y_k) + \nabla f(x_k)^T(y_k - x_k) + \tfrac{1}{2L}\|\nabla f(x_k) - \nabla f(y_k)\|^2 \\ + \tfrac{\mu}{2(1-\frac{\mu}{L})}\|x_k - y_k - \tfrac{1}{L}(\nabla f(x_k) - \nabla f(y_k))\|^2 \leq 0, \end{aligned}$$

- smoothness and strong convexity between $y_{k+1}$ and $x_*$ with multiplier $\lambda_2 = 1 - \rho$:

$$\begin{aligned} f(y_{k+1}) - f_* + \nabla f(y_{k+1})^T(x_* - y_{k+1}) + \tfrac{1}{2L}\|\nabla f(y_{k+1})\|^2 \\ + \tfrac{\mu}{2(1-\frac{\mu}{L})}\|y_{k+1} - x_* - \tfrac{1}{L}\nabla f(y_{k+1})\|^2 \leq 0, \end{aligned}$$

- smoothness and strong convexity between $y_{k+1}$ and $x_k$ with multiplier $\lambda_3 = \rho$:

$$\begin{aligned} f(y_{k+1}) - f(x_k) + \nabla f(y_{k+1})^T(x_k - y_{k+1}) + \tfrac{1}{2L}\|\nabla f(y_{k+1}) - \nabla f(x_k)\|^2 \\ + \tfrac{\mu}{2(1-\frac{\mu}{L})}\|y_{k+1} - x_k - \tfrac{1}{L}(\nabla f(y_{k+1}) - \nabla f(x_k))\|^2 \leq 0. \end{aligned}$$

Given that $\lambda_1, \lambda_2, \lambda_3 \geq 0$, the following weighted sum is a valid inequality

$$\begin{aligned} 0 \geq \lambda_1 \Big[ & f(x_k) - f(y_k) + \nabla f(x_k)^T(y_k - x_k) + \tfrac{1}{2L}\|\nabla f(x_k) - \nabla f(y_k)\|^2 \\ & + \tfrac{\mu}{2(1-\frac{\mu}{L})}\|x_k - y_k - \tfrac{1}{L}(\nabla f(x_k) - \nabla f(y_k))\|^2 \Big] \end{aligned}$$

$$
\begin{aligned}
&+ \lambda_2 \Big[ f(y_{k+1}) - f_* + \nabla f(y_{k+1})^T (x_* - y_{k+1}) + \tfrac{1}{2L} \|\nabla f(y_{k+1})\|^2 \\
&\quad + \tfrac{\mu}{2(1-\frac{\mu}{L})} \|y_{k+1} - x_* - \tfrac{1}{L} \nabla f(y_{k+1})\|^2 \Big] \\
&+ \lambda_3 \Big[ f(y_{k+1}) - f(x_k) + \nabla f(y_{k+1})^T (x_k - y_{k+1}) + \tfrac{1}{2L} \|\nabla f(y_{k+1}) - \nabla f(x_k)\|^2 \\
&\quad + \tfrac{\mu}{2(1-\frac{\mu}{L})} \|y_{k+1} - x_k - \tfrac{1}{L} (\nabla f(y_{k+1}) - \nabla f(x_k))\|^2 \Big],
\end{aligned}
$$

which can be reformulated exactly, using the notation

$$
\begin{aligned}
\Phi(x, y) &= f(y) - f_* + \tfrac{L-\mu}{2} \|x - y\|^2 \\
y_{k+1} &= x_k - \tfrac{1}{L} \nabla f(x_k) \\
x_{k+1} &= y_{k+1} + \beta_k (y_{k+1} - y_k)
\end{aligned}
$$

along with the expression of $\rho$, in the form

$$
\begin{aligned}
0 \geq & \Phi(x_{k+1}, y_{k+1}) - \rho \Phi(x_k, y_k) \\
&+ \tfrac{1}{2(L-\mu)} \left\| (1-\rho)(\nabla f(x_k) - L(x_k - x_*)) + \nabla f(y_{k+1}) \right\|^2 \\
&+ \tfrac{\rho}{2(L-\mu)} \left\| \nabla f(y_k) - \nabla f(x_k) + \mu (x_k - y_k) \right\|^2 \\
&+ \tfrac{(1-\beta^2)\rho}{2L} \|\nabla f(x_k) + L(y_k - x_k)\|^2.
\end{aligned}
$$

Therefore, using the assumption $\beta_k \in [0, 1]$, we finally arrive to the desired

$$
\Phi(x_{k+1}, y_{k+1}) \leq \rho \Phi(x_k, y_k).
$$

■

We then get the following corollary on the primal gap after $N$ iterations.

**Corollary 4.3.2.** *Let $f \in \mathcal{F}_{\mu,L}$, a number of iterations $N \in \mathbb{N}$, and consider Algorithm 4.3 with a sequence $\{\beta_k\}_k$ satisfying $\beta_k \in [0, 1]$ for all $k \in [1, N]$. Then, for any $x_0 \in \mathbb{R}^d$, it holds that*

$$
f(y_N) - f_* \leq \left(1 - \frac{\mu}{L}\right)^N (f(x_0) - f_*),
$$

*where $f_* = \min_x f(x)$.*

*Proof.* Using the potential function argument of Lemma 4.3.1, we can nested together the inequalities and get

$$
f(y_N) - f_* \leq \Phi(x_N, y_N) \leq \ldots \leq \rho^N \Phi(x_0, y_0) = f(x_0) - f_*,
$$

where we used $x_0 = y_0$ in the last equality. ■

This result shows the robustness of AGM with respect to the momentum parameter. Adaptive strategies, that modify the momentum term in the algorithm automatically, thus at least enjoy the gradient method's convergence rate when $\beta_k$ is kept within the interval $[0, 1]$—this is the case for both (Acc. Variant I) and (Acc. Variant II). To our knowledge, only non-blowup properties [Lin and Xiao, 2014, Lemma 1] were known when overestimating $\mu$.

### 4.3.2 Polyak steps based momentum

The momentum term in (Acc. Variant I) was designed using the inverse of Polyak step as an estimate of the strong convexity parameter. This choice of strong convexity estimate is motivated by the fact that under some mild assumptions on $f$ (i.e., for quadratic or self-concordant $f$), the quantity $\|\nabla f(z_k)\|^2/(2(f(z_k)-f_*))$ converges to the strong convexity constant at optimum when the $z_k$ are iterates of gradient descent algorithm with step size $1/L$. Indeed, when $f$ is e.g. quadratic, Polyak steps can be expressed as

$$\begin{aligned}\frac{\|\nabla f(z_k)\|^2}{2(f(z_k)-f_*)} &= \frac{\|\nabla^2 f(z_k-x_*)\|^2}{\langle z_k-x_*, \nabla^2 f(z_k-x_*)\rangle} \\ &= \frac{\sum_{i=1}^d \nu_i^2 \langle z_k-x_*, e_i\rangle^2}{\sum_{i=1}^d \nu_i \langle z_k-x_*, e_i\rangle^2},\end{aligned}$$

where $x_* \in \operatorname{argmin}_x f(x)$, $0<\mu=\nu_1\leq\ldots\leq\nu_d=L$ the eigenvalues of $\nabla^2 f$ and $\{e_i\}_{i=1,\ldots,d}$ an orthonormal basis of eigenvectors. When $\{z_k\}_k$ is obtained through a gradient descent with step size $\frac{1}{L}$, the residuals $z_k-x_*$ tend to align with eigenvectors associated to the smallest eigenvalue $\nu_1=\mu$, and therefore the inverse Polyak steps converge to $\mu$.

In order for $\tilde{\mu}_k$ to be always defined and within the interval $[\mu, L]$, we assume that iterates never reach exactly optimality. Under this condition we have $\beta_k \in [0, \beta_*]$ and Corollary 4.3.2 readily applies to both (Acc. Variant I) or (Acc. Variant II). However, this result can be improved for those particular choices, as described in Lemma 4.3.3 and Proposition 4.3.4, as the rate can be expressed in terms of the local $\tilde{\mu}_k$ instead of $\mu$.

**Lemma 4.3.3** (Adaptivity)**.** *Let $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$, some iteration number $k \in \mathbb{N}$, and consider Algorithm 4.3 with either* (Acc. Variant I) *or* (Acc. Variant II)*. For any $x_k, y_k \in \mathbb{R}^d$ such that $\tilde{\mu}_k$ well defined, it holds that*

$$\Phi(x_{k+1}, y_{k+1}) \leq \rho(\tilde{\mu}_k)\Phi(x_k, y_k) \tag{4.7}$$

*where $\Phi(x,y)=\frac{L}{2}\|x-y\|^2+f(y)-f_*$, $f_*=\min_x f(x)$ and $\rho(\tilde{\mu})=\frac{1}{1+\frac{\tilde{\mu}}{L}}$. Otherwise $\nabla f(y_{k+1})=0$.*

*Proof.* We denote by $\rho(x)$ the function $\rho(x)=\frac{1}{1+\frac{x}{L}}$. The proof consists in the following combination of inequalities obtained from Lemma 4.1.1:

- smoothness and convexity between $y_{k+1}$ and $x_k$ with multiplier $\lambda_1=\rho(\tilde{\mu}_k)$:

$$f(y_{k+1})-f(x_k)+\nabla f(y_{k+1})^T(x_k-y_{k+1})+\tfrac{1}{2L}\|\nabla f(x_k)-\nabla f(y_{k+1})\|^2\leq 0,$$

- convexity between $x_k$ and $y_k$ with multiplier $\lambda_2=\rho(\tilde{\mu}_k)$:

$$f(x_k)-f(y_k)+\nabla f(x_k)^T(y_k-x_k)\leq 0,$$

- definition of $\tilde{\mu}_k$ with multiplier $\lambda_3=\frac{1-\rho(\tilde{\mu}_k)}{2\tilde{\mu}_k}$:

$$2\tilde{\mu}_k(f(y_{k+1})-f_*)-\|\nabla f(y_{k+1})\|^2\leq 0$$

(we use an inequality so that it also holds for $\tilde{\mu}_k=\min\{\tilde{\mu}_{k-1}, \frac{\|\nabla f(y_{k+1})\|^2}{2(f(y_{k+1})-f_*)}\}$).

The following weighted sum is a valid inequality given that $\lambda_1, \lambda_2, \lambda_3 \geq 0$:

$$\begin{aligned}0 \geq &\lambda_1 \left[f(y_{k+1}) - f(x_k) + \nabla f(y_{k+1})^T (x_k - y_{k+1}) + \tfrac{1}{2L}\|\nabla f(x_k) - \nabla f(y_{k+1})\|^2\right] \\ &+ \lambda_2 \left[f(x_k) - f(y_k) + \nabla f(x_k)^T (y_k - x_k)\right] \\ &+ \lambda_3 \left[2\tilde{\mu}_k (f(y_{k+1}) - f_*) - \|\nabla f(y_{k+1})\|^2\right],\end{aligned}$$

which can be reformulated exactly, using the notation

$$\begin{aligned}\Phi(x, y) &= f(y) - f_* + \tfrac{L}{2}\|x - y\|^2 \\ y_{k+1} &= x_k - \tfrac{1}{L}\nabla f(x_k) \\ x_{k+1} &= y_{k+1} + \beta_k (y_{k+1} - y_k) \\ \beta_k &= \frac{\sqrt{L} - \sqrt{\tilde{\mu}_k}}{\sqrt{L} + \sqrt{\tilde{\mu}_k}}\end{aligned}$$

along with the expression for $\rho(x)$, in the form

$$\begin{aligned}0 \geq &\Phi(x_{k+1}, y_{k+1}) - \rho(\tilde{\mu}_k)\Phi(x_k, y_k) \\ &+ \frac{\left(4L^2\sqrt{\frac{\tilde{\mu}_k}{L}} - L\left(\tilde{\mu}_k - 2\tilde{\mu}_k\sqrt{\frac{\tilde{\mu}_k}{L}}\right) - \tilde{\mu}_k^2\right)}{2L^2(L+\tilde{\mu}_k)\left(\sqrt{\frac{\tilde{\mu}_k}{L}}+1\right)^2}\|\nabla f(x_k) + L(y_k - x_k)\|^2,\end{aligned}$$

which, in turns, is equivalent to

$$\begin{aligned}\Phi(x_{k+1}, y_{k+1}) \leq &\rho(\tilde{\mu}_k)\Phi(x_k, y_k) - \frac{\left(4L^2\sqrt{\frac{\tilde{\mu}_k}{L}} - L\left(\tilde{\mu}_k - 2\tilde{\mu}_k\sqrt{\frac{\tilde{\mu}_k}{L}}\right) - \tilde{\mu}_k^2\right)}{2L^2(L+\tilde{\mu}_k)\left(\sqrt{\frac{\tilde{\mu}_k}{L}}+1\right)^2}\|\nabla f(x_k) + L(y_k - x_k)\|^2, \\ \leq &\rho(\tilde{\mu}_k)\Phi(x_k, y_k)\end{aligned}$$

where the inequality follows from the sign of the term we removed, so it remains to show that

$$4L^2\sqrt{\tfrac{\tilde{\mu}_k}{L}} - L\left(\tilde{\mu}_k - 2\tilde{\mu}_k\sqrt{\tfrac{\tilde{\mu}_k}{L}}\right) - \tilde{\mu}_k^2 \geq 0 \quad \forall \tilde{\mu}_k \in [0, L].$$

Indeed, evaluating the sign of the previous expression boils down to study that of $g(x) = 4\sqrt{x} - (x - 2x\sqrt{x}) - x^2$ on $[0, 1]$, which follows from:

$$g(x) \geq 3\sqrt{x} + x\sqrt{x} \geq 0 \quad \forall x \in [0, 1].$$

■

**Proposition 4.3.4.** *Let $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$, some number of iterations $N \in \mathbb{N}$, and consider Algorithm 4.3 with either (Acc. Variant I) or (Acc. Variant II). Then, for any $x_0 \in \mathbb{R}^d$, such that the sequence $\{\tilde{\mu}_k\}_k$ is well defined , it holds that*

$$f(y_N) - f_* \leq \left(\prod_{k=0}^{N-1} \rho(\tilde{\mu}_k)\right)(f(x_0) - f_*)$$

*where $f_* = \min_x f(x)$ and $\rho(\tilde{\mu}) = \frac{1}{1+\frac{\tilde{\mu}}{L}}$. Otherwise $\nabla f(y_k) = 0$ with $k \in [0, N]$.*

*Proof.* Use Lemma 4.3.3 recursively and notice that $\Phi(x_0, y_0) = f(x_0) - f_*$. ■

In fact, these results on (Acc. Variant I) and (Acc. Variant II) also hold under Hölderian error bounds [Bolte et al., 2007, 2017] (also known as Kurdyka-Łojasewicz, Polyak-Łojasewicz, quadratic growth, etc.) which require the existence of $\mu > 0$ such that for all $x \in \mathbb{R}^d$, $f(x) - f_* \leq \frac{1}{2\mu}\|\nabla f(x)\|^2$. This condition holds in particular for strongly convex function but is much weaker.

**Corollary 4.3.5.** *Under the conditions of Proposition 4.3.4, if there exists $\mu > 0$ such that for all $x \in \mathbb{R}^d$, $f(x) - f_* \leq \frac{1}{2\mu}\|\nabla f(x)\|^2$ then after $N \in \mathbb{N}$ iterations*

$$f(y_N) - f_* \leq \left(1 + \frac{\mu}{L}\right)^{-N} (f(x_0) - f_*),$$

*where $f_* = \min_x f(x)$.*

Looking at Proposition 4.3.4 more closely, we notice that when the estimates $\tilde{\mu}_k$ are larger than $\sqrt{L\mu}$, the adaptive accelerated method exhibits an accelerated linear convergence rate $O(1 - \sqrt{\frac{\mu}{L}})$. It remains to study the convergence of the adaptive method in the regime where $\tilde{\mu}_k$ is small. In this case, we provide another robustness result for the AGM algorithm when the momentum $\beta_k$ is close enough to its classical value (Const-mom).

**Lemma 4.3.6.** *Let $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, some iteration number $k \in \mathbb{N}$, and consider Algorithm 4.3 with*

$$\frac{\sqrt{L} - \sqrt[4]{L\mu}}{\sqrt{L} + \sqrt[4]{L\mu}} \leq \beta_k \leq \beta_* = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}.$$

*Then, for any $x_k, y_k \in \mathbb{R}^d$, it holds that*

$$\Phi(x_{k+1}, y_{k+1}) \leq \rho \Phi(x_k, y_k) \tag{4.8}$$

*where $\Phi(x, y) = \frac{L}{2}\|\frac{1}{\sqrt{\rho}}(x - x_*) - \sqrt{\rho}(y - x_*)\|^2 + f(y) - f_*$, $x_* \in \operatorname{argmin}_x f(x)$, $f_* = f(x_*)$ and $\rho = \left(1 + \left(\frac{\mu}{L}\right)^{\frac{3}{4}}\right)^{-1}$.*

*Proof.* The proof, that follows the same steps as the previous ones, but with some additional technicalities for certifying the positivity of some polynomials, is deferred to Section 4.A. ■

This lemma guarantees a linear convergence rate $O\left(1 - \left(\frac{\mu}{L}\right)^{3/4}\right)^k$ that is slower than the accelerated rate with full knowledge of $\mu$ but faster than the gradient rate. We now combine the convergence results for the two regimes of $\tilde{\mu}_k$, and get a global linear convergence rate for (Acc. Variant II).

**Proposition 4.3.7** (Adaptive AGM). *Let $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, and $N \in \mathbb{N}$ be a number of iterations. We consider Algorithm 4.3 with (Acc. Variant II), and let $\{y_k, x_k\}_k$ be the iterates of the method. Then, for any $x_0 \in \mathbb{R}^d$, such that the sequence $\{\tilde{\mu}_k\}_k$ is well defined, we let $m \in N$ be the first integer such that $\frac{\|\nabla f(y_{m+1})\|^2}{2(f(y_{m+1}) - f_*)} \leq \sqrt{L\mu}$, (let $m = \infty$ if this never happens during the $N$ iterations),*

$$f(y_N) - f_* \leq \begin{cases} \rho_1^N \left(\frac{L}{2}\left(\frac{1}{\sqrt{\rho_1}} - \sqrt{\rho_1}\right)^2 \|x_0 - x_*\|^2 + f(x_0) - f_*\right) & \text{if } m = 0, \\ \rho_2^N (f(x_0) - f_*) & \text{if } m = \infty, \\ C\rho_1^{N-m}\rho_2^m (f(x_0) - f_*) & \text{otherwise,} \end{cases}$$

*where* $x_* \in \operatorname{argmin}_x f(x)$, $f_* = f(x_*)$, $C = \left(\left(\frac{1}{\rho_1} - 1\right)\left(1 + \sqrt{\frac{L}{2\mu}}\right)^2 + 1\right)$, $\rho_1 = \left(1 + \left(\frac{\mu}{L}\right)^{\frac{3}{4}}\right)^{-1}$ *and* $\rho_2 = \left(1 + \sqrt{\frac{\mu}{L}}\right)^{-1}$.
*Otherwise* $\nabla f(y_k) = 0$ *with* $k \in [0, N]$.

*Proof.* The case $m = 0$ results from Lemma 4.3.6 applied recursively and the case $m = \infty$ result from Proposition 4.3.4. In the following we consider that $m \in [1, N]$. Then for $(y_k, x_k)_{k\in[m+1,N]}$,

$$\frac{\sqrt{L} - \sqrt[4]{L\mu}}{\sqrt{L} + \sqrt[4]{L\mu}} \leq \beta_{k-1} \leq \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}$$

and Lemma 4.3.6 applies so

$$f(y_N) - f_* \leq \rho_1^{N-m}\left(\frac{L}{2}\|\frac{1}{\sqrt{\rho_1}}(x_m - x_*) - \sqrt{\rho_1}(y_m - x_*)\|^2 + f(y_m) - f_*\right)$$

and we have

$$\begin{aligned}
&\frac{L}{2}\|\frac{1}{\sqrt{\rho_1}}(x_m - x_*) - \sqrt{\rho_1}(y_m - x_*)\|^2 + f(y_m) - f_* \\
&= \frac{L}{2}\left(\frac{1}{\rho_1} - 1\right)\|x_m - x_*\|^2 - \frac{L}{2}(1 - \rho_1)\|y_m - x_*\|^2 + \frac{L}{2}\|x_m - y_m\|^2 + f(y_m) - f_* \\
&\leq \frac{L}{2}\left(\frac{1}{\rho_1} - 1\right)\|x_m - x_*\|^2 + \frac{L}{2}\|x_m - y_m\|^2 + f(y_m) - f_* \\
&\leq \frac{L}{2}\left(\frac{1}{\rho_1} - 1\right)(\|x_m - y_m\| + \|y_m - x_*\|)^2 + \frac{L}{2}\|x_m - y_m\|^2 + f(y_m) - f_* \\
&\leq \left(\frac{1}{\rho_1} - 1\right)\left(\sqrt{\frac{L}{2}}\|x_m - y_m\| + \sqrt{\frac{L}{2\mu}}\sqrt{f(y_m) - f_*}\right)^2 + \frac{L}{2}\|x_m - y_m\|^2 + f(y_m) - f_*
\end{aligned}$$

We can now apply Corollary 4.3.5. From the definition of $m$, we have

$$2(f(y_k) - f_*) \leq \frac{1}{\sqrt{L\mu}}\|\nabla f(y_k)\|^2 \text{ for all } k \in [1, m].$$

Therefore, by denoting $\rho_2 = \left(1 + \sqrt{\frac{\mu}{L}}\right)^{-1}$, we have the following inequalities

$$\begin{aligned}
\frac{L}{2}\|x_m - y_m\|^2 + f(y_m) - f_* &\leq \rho_2^m(f(x_0) - f_*), \\
\sqrt{\frac{L}{2}}\|x_m - y_m\| &\leq \rho_2^{m/2}\sqrt{f(x_0) - f_*}, \\
\sqrt{\frac{L}{2\mu}}\sqrt{f(y_m) - f_*} &\leq \sqrt{\frac{L}{2\mu}}\rho_2^{m/2}\sqrt{f(x_0) - f_*},
\end{aligned}$$

which leads to

$$\frac{L}{2}\|\frac{1}{\sqrt{\rho_1}}(x_m - x_*) - \sqrt{\rho_1}(y_m - x_*)\|^2 + f(y_m) - f_*$$

$$\leq \left( \left( \frac{1}{\rho_1} - 1 \right) \left( 1 + \sqrt{\frac{L}{2\mu}} \right)^2 + 1 \right) \rho_2^m (f(x_0) - f_*),$$

reaching the desired result. ■

The previous convergence bound is only valid for (Acc. Variant II) mostly for technical reasons. Indeed the min is present in order to have at most one transition between the regime $\tilde{\mu}_k \geq \sqrt{L\mu}$ and $\tilde{\mu}_k \leq \sqrt{L\mu}$. In practice, however, we didn't observe any difference between the behaviors of (Acc. Variant I) and that of (Acc. Variant II).

In the following, we present an (unpractical) way of removing the need for $f_*$.

### 4.3.3 Removing the dependence on the optimal value

The key ingredients in Proposition 4.3.7 are, (i) the fact that the method is fast when $\tilde{\mu}$ is large, (ii) the robustness of AGM when $\tilde{\mu}$ is small. In the following, we describe an algorithm together with estimates of $\mu$ that follows the same arguments.

**Accelerated gradient method (AGM) no $f_*$** (Algorithm 4.3.3)

**Input:**

- Objective function: $f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$.
- Initial guess: $x_0 \in \mathbb{R}^d$.
- Smoothness parameter: $L \in \mathbb{R}^*$.
- Number of iterations: $N \in \mathbb{N}^*$.

**Initialization:**

$\tilde{\mu}_0 = L$

**Run:**

For $k = 0, \ldots, N-1$:

$$x_{k+1} = x_k - \tfrac{1}{L}\nabla f(x_k)$$
$$\tilde{\mu}_{k+1} = \min\left(\tilde{\mu}_k, L\left(1 - \frac{\|\nabla f(x_{k+1})\|}{\|\nabla f(x_k)\|}\right)\right)$$

End For

$z_0 = x_N,\ y_0 = z_0,\ \beta = \frac{\sqrt{L}-\sqrt{\tilde{\mu}_N}}{\sqrt{L}+\sqrt{\tilde{\mu}_N}}$

For $k = 0, \ldots, N-1$:

$$y_{k+1} = z_k - \tfrac{1}{L}\nabla f(z_k)$$
$$z_{k+1} = y_{k+1} + \beta(y_{k+1} - y_k)$$

End For

**Output:** $y_N$

The following lemma ensures that the estimates of $\mu$ used in Algorithm 4.3.3 stay in $[\mu, L]$.

**Lemma 4.3.8.** *Let $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, $x_k \in \mathbb{R}^d$ and $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$. It holds that*

$$\mu \leq L\left(1 - \frac{\|\nabla f(x_{k+1})\|}{\|\nabla f(x_k)\|}\right) \leq L.$$

*Proof.* The right hand side inequality is direct. For the left hand one, we add the two inequalities from Lemma 4.1.1

- smoothness and strong convexity between $x_{k+1}$ and $x_k$:

$$\begin{aligned} f(x_{k+1}) - f(x_k) + \langle \nabla f(x_{k+1}), x_k - x_{k+1}\rangle + \tfrac{1}{2L}\|\nabla f(x_k) - \nabla f(x_{k+1})\|^2 \\ + \tfrac{\mu}{2(1-\frac{\mu}{L})}\|x_k - x_{k+1} - \tfrac{1}{L}(\nabla f(x_k) - \nabla f(x_{k+1}))\|^2 \leq 0, \end{aligned}$$

- smoothness and strong convexity between $x_k$ and $x_{k+1}$:

$$\begin{aligned} f(x_k) - f(x_{k+1}) + \langle \nabla f(x_k), x_{k+1} - x_k\rangle + \tfrac{1}{2L}\|\nabla f(x_k) - \nabla f(x_{k+1})\|^2 \\ + \tfrac{\mu}{2(1-\frac{\mu}{L})}\|x_k - x_{k+1} - \tfrac{1}{L}(\nabla f(x_k) - \nabla f(x_{k+1}))\|^2 \leq 0. \end{aligned}$$

Summing these two inequalities gives

$$\begin{aligned} \langle \nabla f(x_{k+1}) - \nabla f(x_k), x_k - x_{k+1}\rangle + \tfrac{1}{L}\|\nabla f(x_k) - \nabla f(x_{k+1})\|^2 \\ + \tfrac{\mu}{(1-\frac{\mu}{L})}\|x_k - x_{k+1} - \tfrac{1}{L}(\nabla f(x_k) - \nabla f(x_{k+1}))\|^2 \leq 0, \end{aligned}$$

and substituting the expression of $x_{k+1}$ leads to

$$\tfrac{1}{L}\langle \nabla f(x_{k+1}) - \nabla f(x_k), \nabla f(x_k)\rangle + \tfrac{1}{L}\|\nabla f(x_k) - \nabla f(x_{k+1})\|^2 + \tfrac{\mu}{L(L-\mu)}\|\nabla f(x_{k+1})\|^2 \leq 0,$$

which can be reformulated as

$$\tfrac{1}{L}\left(1 + \tfrac{\mu}{(L-\mu)}\right)\|\nabla f(x_{k+1})\|^2 \leq \tfrac{1}{L}\langle \nabla f(x_{k+1}), \nabla f(x_k)\rangle.$$

Using Cauchy-Schwarz in the left hand side, we finally obtain

$$\|\nabla f(x_{k+1})\| \leq (1 - \tfrac{\mu}{L})\,\|\nabla f(x_k)\|,$$

from which the desired conclusion follows easily. ■

**Proposition 4.3.9.** *Let $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, $N \in \mathbb{N}^*$ be a number of iterations. Denote by $\{x_k, y_k, z_k\}_k$ the iterates of Algorithm 4.3.3 initiated at $x_0 \in \mathbb{R}^d$. It holds that*

$$f(y_N) - f_* \leq \begin{cases} \rho_1^N \rho_2^{2N}\left(\frac{L}{2}\left(\frac{1}{\sqrt{\rho_1}} - \sqrt{\rho_1}\right)^2\|x_0 - x_*\|^2 + f(x_0) - f_*\right) & \text{if } \tilde{\mu}_N \leq \sqrt{L\mu}, \\ \frac{1}{2\mu}\rho_2^N \rho_3^{2N}\|\nabla f(x_0)\|^2 & \text{otherwise,} \end{cases}$$

*where $x_* \in \operatorname{argmin}_x f(x)$, $f_* = f(x_*)$, $\rho_1 = \left(1 + (\frac{\mu}{L})^{3/4}\right)^{-1}$, $\rho_2 = (1 - \frac{\mu}{L})$ and $\rho_3 = \left(1 - \sqrt{\frac{\mu}{L}}\right)$.*

*Proof.* We treat the two regimes separately.

- **First case :** $\tilde{\mu}_N \leq \sqrt{L\mu}$.
  Combining inequalities of Lemma 4.3.6 leads to

$$f(y_N) - f_* \leq \rho_1^N \left( \frac{L}{2} \left( \frac{1}{\sqrt{\rho_1}} - \sqrt{\rho_1} \right)^2 \|x_N - x_*\|^2 + f(x_N) - f_* \right),$$

  where $\rho_1 = \left(1 + \left(\frac{\mu}{L}\right)^{3/4}\right)^{-1}$. Then, we use that gradient descent with step size $\frac{1}{L}$ makes squared distance to optimum and function values decrease linearly with a rate $\rho_2^2 = \left(1 - \frac{\mu}{L}\right)^2$ (see e.g. Taylor et al. [2018a, Theorem 2.1] with $h = 0$). This finally produces the bound

$$f(y_N) - f_* \leq \rho_1^N \rho_2^{2N} \left( \frac{L}{2} \left( \frac{1}{\sqrt{\rho_1}} - \sqrt{\rho_1} \right)^2 \|x_0 - x_*\|^2 + f(x_0) - f_* \right)$$

- **Second case :** $\tilde{\mu}_N \geq \sqrt{L\mu}$.
  Using Corollary 4.3.2 we can write

$$f(y_N) - f_* \leq \rho_2^N (f(x_N) - f_*),$$

  and using strong convexity of the objective, we obtain

$$f(y_N) - f_* \leq \tfrac{1}{2\mu} \rho_2^N \|\nabla f(x_N)\|^2. \tag{4.9}$$

  Finally, we get from the definition of the $\tilde{\mu}_k$'s that

$$\begin{aligned} \|\nabla f(x_N)\| &\leq \prod_{k=1}^{N} \left(1 - \tfrac{\tilde{\mu}_k}{L}\right) \|\nabla f(x_0)\| \\ &\leq \left(1 - \tfrac{\tilde{\mu}_N}{L}\right)^N \|\nabla f(x_0)\|, \end{aligned} \tag{4.10}$$

  and re-injecting the bound (4.10) in (4.9) together with the condition $\tilde{\mu}_N \geq \sqrt{L\mu}$ give the desired result.

■

Proposition 4.3.9 guarantees that Algorithm 4.3.3 converges at a linear rate $O\left(1 - \frac{1}{2}\left(\frac{\mu}{L}\right)^{3/4}\right)$ when $\mu/L \ll 1$. This method illustrates how the robustness results on AGM can be used to provide accelerated algorithms. However, note that Algorithm 4.3.3 is not really practical as the gradient descent steps can be very slow.

## 4.4 Analysis mechanisms

Starting with the work of Drori and Teboulle [2014], computer-aided worst-case analyses of convex optimization methods have provided a generic technique producing convergence rates for many classical first-order algorithms. The results in [Drori and Teboulle, 2014, Taylor et al., 2017c] use an interpolation argument to write the problem of finding the worst case behavior of an algorithm, given a convergence criterion, as a tractable semidefinite program—often referred to as a Performance Estimation Problem (PEP). We adapted the technique for generating the complexity bounds on gradient methods with Polyak steps.

Our proofs were obtained by searching for Lyapunov (or potential) functions (see e.g. [Bansal and Gupta, 2019] for a recent survey) as briefly presented in Section 3.2.3 of Chapter 3. We also refer the reader to the discussions on PEPs in [Taylor and Bach, 2019, Taylor et al., 2018b] for more details. A related line of works (equivalent in many situations) is that of integral quadratic constraints [Lessard et al., 2016], which leverage results from control theory to perform worst-case complexity analysis. All these approaches were originally developed for non adaptive methods and in what follows, we show how we used the PEP approach for adaptive algorithms. A similar reasoning would allow adapting IQCs for adaptive methods as well.

To fix ideas and illustrate our procedure, we first analyze the worst case complexity of a variant of the classical gradient method with Polyak steps, and show improved convergence bounds compared to classical results (see Hazan and Kakade [2019] for a recent treatment). We consider the gradient method with Polyak steps described in Algorithm 4.2 with (Variant I) for $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$. Notice that there is a factor two in the step size that is not present in the original Polyak step. This factor simplifies, and improves, the analysis for the convergence in terms of distance to the optimum.

To prove a linear convergence rate, we can focus on the improvement yielded by a single iteration of the form

$$x_{k+1} := x_k - \gamma_k \nabla f(x_k), \quad \text{where} \quad \gamma_k := 2\frac{f(x_k) - f_*}{\|\nabla f(x_k)\|^2}. \tag{4.11}$$

We seek to bound the worst case (i.e., smallest) decrease in $\|x_{k+1} - x_*\|^2$ relative to $\|x_k - x_*\|^2$ when $x_{k+1}$ is obtained using the iteration in (4.11) for any function $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ and any point $x_k$. In other words we seek to solve the following optimization problem

$$\begin{array}{ll} \text{maximize} & \dfrac{\|x_{k+1} - x_*\|^2}{\|x_k - x_*\|^2} \\ \text{subject to} & x_{k+1} = x_k - 2\frac{f(x_k)-f_*}{\|\nabla f(x_k)\|^2}\nabla f(x_k), \\ & f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d),\ x_k \in \mathbb{R}^d,\ d \in \mathbb{N}. \end{array} \tag{4.12}$$

in the variables $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ and $x_k, x_{k+1}, x_*, \nabla f(x_k) \in \mathbb{R}^d$, with parameter $f_* \in \mathbb{R}$. The key argument in [Drori and Teboulle, 2014, Taylor et al., 2017c] is that the constraint on the regularity of the function $f$ in problem (4.12) can be replaced by a finite number of inequalities from Lemma 4.1.1. We get an upper bound on the optimum of problem (4.12) by relaxing the constraint $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, keeping just two inequalities from Lemma 4.1.1 relating $x_k$ and $x_*$ to obtain the following relaxed problem

$$\begin{array}{ll} \text{maximize} & \dfrac{\|x_{k+1} - x_*\|^2}{\|x_k - x_*\|^2} \\ \text{subject to} & f_k - f_* + g_k^T(x_* - x_k) + \frac{1}{2L}\|g_k\|^2 + \frac{\mu}{2(1-\frac{\mu}{L})}\|x_k - x_* - \frac{1}{L}g_k\|^2 \leq 0 \\ & f_* - f_k + \frac{1}{2L}\|g_k\|^2 + \frac{\mu}{2(1-\frac{\mu}{L})}\|x_k - x_* - \frac{1}{L}g_k\|^2 \leq 0 \\ & x_{k+1} = x_k - 2\frac{f_k - f_*}{\|g_k\|^2}g_k \end{array} \tag{4.13}$$

in the variables $x_k, x_*, g_k \in \mathbb{R}^d$ and $f_k, f_* \in \mathbb{R}$.

This relaxed problem is finite dimensional, but still depends on the dimension of the ambient space while we are interested in convergence rates independent of the dimension. One of the key insights of the PEP approach is to notice that (4.13) can be kernelized, i.e., written in terms of the quadratic variables $X_k = \|x_k - x_*\|^2$, $G_k = \|g_k\|^2$, $GX_k = g_k^T(x_* - x_k)$ in addition to $f_k$ and $f_*$. Indeed,

problem (4.13) is equivalent to solving

$$\begin{array}{ll} \text{maximize} & 1 + 4\frac{f_k - f_*}{G_k}\frac{GX_k}{X_k} + 4\frac{(f_k-f_*)^2}{G_k X_k} \\ \text{subject to} & f_k - f_* + GX_k + \frac{1}{2L}G_k + \frac{\mu}{2(1-\frac{\mu}{L})}\left(X_k + \frac{2}{L}GX_k + \frac{1}{L^2}G_k\right) \leq 0 \\ & f_* - f_k + \frac{1}{2L}G_k + \frac{\mu}{2(1-\frac{\mu}{L})}\left(X_k + \frac{2}{L}GX_k + \frac{1}{L^2}G_k\right) \leq 0 \\ & \begin{pmatrix} X_k & GX_k \\ GX_k & G_k \end{pmatrix} \succcurlyeq 0 \end{array} \tag{4.14}$$

in the variables $X_k, G_k, GX_k, f_k, f_* \in \mathbb{R}$. This new problem has only five real variables but is not readily tractable because of the non-linearity in the objective. By homogeneity we can impose $X_k = 1$ without loss of generality. We introduce a step size variable $\gamma$ to rewrite the problem as

$$\begin{array}{ll} \text{maximize} & \rho(\gamma) \\ \text{subject to} & \gamma \in \mathbb{R} \end{array} \tag{4.15}$$

where

$$\begin{array}{lll} \rho(\gamma) := & \text{max.} & 1 + 2\gamma GX_k + 2(f_k - f_*)\gamma \\ & \text{s.t.} & f_k - f_* + GX_k + \frac{1}{2L}G_k + \frac{\mu}{2(1-\frac{\mu}{L})}\left(X_k + \frac{2}{L}GX_k + \frac{1}{L^2}G_k\right) \leq 0 \\ & & f_* - f_k + \frac{1}{2L}G_k + \frac{\mu}{2(1-\frac{\mu}{L})}\left(X_k + \frac{2}{L}GX_k + \frac{1}{L^2}G_k\right) \leq 0 \\ & & \begin{pmatrix} X_k & GX_k \\ GX_k & G_k \end{pmatrix} \succcurlyeq 0 \\ & & X_k = 1,\ G_k\gamma = 2(f_k - f_*) \end{array} \tag{4.16}$$

which is a semidefinite program. Given $\gamma$, $\rho(\gamma)$ can thus be computed efficiently and our relaxation upper bound on the convergence rate of the method is then given by the maximum value of $\rho(\gamma)$. Note that due to the definition of the step size, we only need to study $\rho(\gamma)$ on the interval $[\frac{1}{L}, \frac{1}{\mu}]$. Figure 4.2 (left) plots $\rho(\gamma)$ for fixed values $\mu = 0.1$ and $L = 1$, and shows (right) the maximum value of $\rho(\gamma)$ for various condition numbers. In this experiment, the worst case convergence rates we obtained numerically appear to perfectly match the bound $(L-\mu)^2/(L+\mu)^2$.

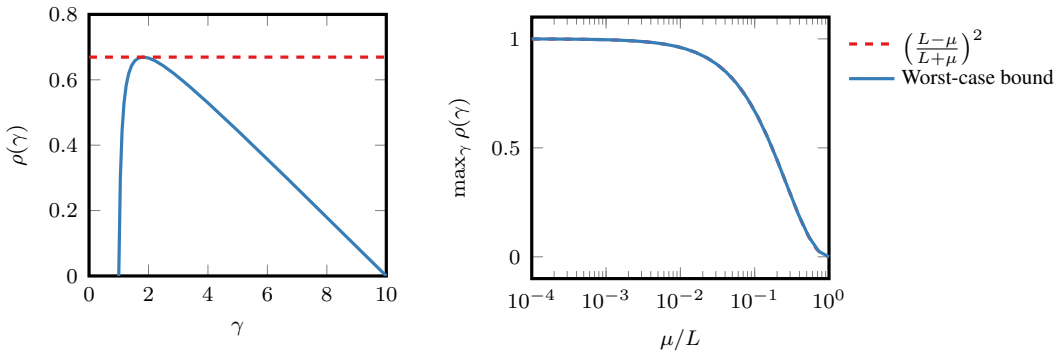


Figure 4.2: Left: we plot $\rho(\gamma)$, by solving (4.16) with $\mu = 0.1$ and $L = 1$. Right: Worst case rate $\max_\gamma \rho(\gamma)$, by solving (4.15), versus inverse condition number.

These numerical observations can in fact be proven analytically as follows. Given a target convergence rate $\rho \in [0, 1]$, we need to show that

$$\|x_{k+1} - x_*\|^2 - \rho\|x_k - x_*\|^2 \leq 0 \tag{4.17}$$

for all *feasible* values of $x_k, x_{k+1}, x_* \in \mathbb{R}^d$, satisfying the constraints of problem (4.13). In the spirit of the Putinar *positivstellensatz* used in sum of squares solutions of semi-algebraic optimization problems [Putinar, 1993, Lasserre, 2001, Parrilo, 2000], we seek to write a certificate of the validity of inequality (4.17) using a positively weighted sum of valid inequalities satisfied by $x_k, x_{k+1}, x_* \in \mathbb{R}^d$ in (4.13). Here, this means writing

$$\begin{aligned}
&\|x_{k+1} - x_*\|^2 - \rho(\gamma_k)\|x_k - x_*\|^2 = \\
&\lambda_1 \left[ f(x_k) - f_* + \nabla f(x_k)^T (x_* - x_k) + \tfrac{1}{2L}\|\nabla f(x_k)\|^2 + \tfrac{\mu}{2(1-\frac{\mu}{L})}\|x_k - x_* - \tfrac{1}{L}\nabla f(x_k)\|^2 \right] \\
+&\lambda_2 \left[ f_* - f(x_k) + \tfrac{1}{2L}\|\nabla f(x_k)\|^2 + \tfrac{\mu}{2(1-\frac{\mu}{L})}\|x_k - x_* - \tfrac{1}{L}\nabla f(x_k)\|^2 \right] \\
+&\lambda_3 \left[ 2(f(x_k) - f_*) - \gamma_k \|\nabla f(x_k)\|^2 \right] \\
&\leq 0
\end{aligned}$$

for some $\lambda_1, \lambda_2 \geq 0$, $\lambda_3 \in \mathbb{R}$, and using the fact $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$ by construction. Through symbolic computations, or by trial and error, inferring a target convergence rate from optimal values of the semidefinite program, the proof consists in showing that we can pick

$$\rho(\gamma_k) = \tfrac{(\gamma_k L - 1)(1 - \gamma_k \mu)}{\gamma_k (L+\mu) - 1}, \quad \lambda_1 = \tfrac{2\gamma_k(\gamma_k L - 1)}{\gamma_k(L+\mu)-1}, \quad \lambda_2 = \tfrac{2\gamma_k(1-\gamma_k\mu)}{\gamma_k(L+\mu)-1} \quad \text{and} \quad \lambda_3 = \tfrac{\gamma_k(2-\gamma_k(L+\mu))}{\gamma_k(L+\mu)-1}.$$

In practice, the numerical solution of the semidefinite program in (4.16) giving $\rho(\gamma)$ can be used to greedily narrow down the list of valid inequalities required by the proof.

Note that since (4.15) is a semialgebraic problem, we could have used sum-of-squares techniques to prove the convergence rate. However, the multipliers and the rates are fractions in $\gamma_k$. Since one usually doesn't know in advance the form of the denominators, one needs relatively high degree polynomials in the SOS program. This means this approach suffers from the usual SOS issues of poor conditioning and scaling.

## 4.5 Numerical analyses of adaptive methods

In this section, we use the methodology presented in Section 4.4 for performing worst-case analyses of some adaptive algorithms, numerically.

### 4.5.1 Exact line search

We start by looking at the worst-case guarantees of gradient descent with exact line search. That is, we consider updates of the form

$$\left\{ \begin{array}{rcl} \gamma_* & = & \operatorname{argmin}_{\gamma} f(x_k - \gamma \nabla f(x_k)) \\ x_{k+1} & = & x_k - \gamma_* \nabla f(x_k), \end{array} \right.$$

where $f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$, and $x_k \in \mathbb{R}^d$. From Klerk et al. [2017, Theorem 1.2], we know that iterates of gradient descent with exact line search applied to $f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$ satisfy

$$f(x_{k+1}) - f_* \leq \left(\tfrac{L-\mu}{L+\mu}\right)^2 (f(x_k) - f_*), \text{ for all } k \geq 0.$$

This results were obtained using relaxations in the associated performance estimation problem. It turns out that this relaxation is tight for this particular criterion, but not necessarily when using e.g. distance to optimality.

We can study more precisely worst-case guarantees of this type by looking at bounds on the value of the ratio $(f(x_{k+1}) - f_*)/(f(x_k) - f_*)$ versus the actual step size value $\gamma \in \mathbb{R}$. Using orthogonality between consecutive gradients (i.e. necessary and sufficient optimality condition of the line search procedure),

$$\gamma_* = \operatorname*{argmin}_{\gamma} f(x_k - \gamma \nabla f(x_k)) \iff \langle \nabla f(x_k - \gamma_* \nabla f(x_k)), \nabla f(x_k) \rangle = 0,$$

this corresponds to solve the following problem

$$\begin{aligned} \rho(\gamma) := \quad & \max \quad \frac{f(x_{k+1}) - f_*}{f(x_k) - f_*} \\ & \text{s.t.} \quad x_{k+1} = x_k - \gamma \nabla f(x_k), \\ & \quad \langle \nabla f(x_{k+1}), \nabla f(x_k) \rangle = 0, \\ & \quad f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d),\ x_k \in \mathbb{R}^d,\ d \in \mathbb{N}, \end{aligned} \tag{4.18}$$

for several values of $\gamma$. This maximization problem can be reformulated as a semidefinite program similar to Section 4.4 (following the steps in Chapter 3 and the references therein) and numerical worst-case bounds of gradient descent with exact line search in this context are displayed in Figure 4.3. Note that $\rho(\gamma)$ is zero when $\gamma$ is outside the ranges displayed in Figure 4.3.

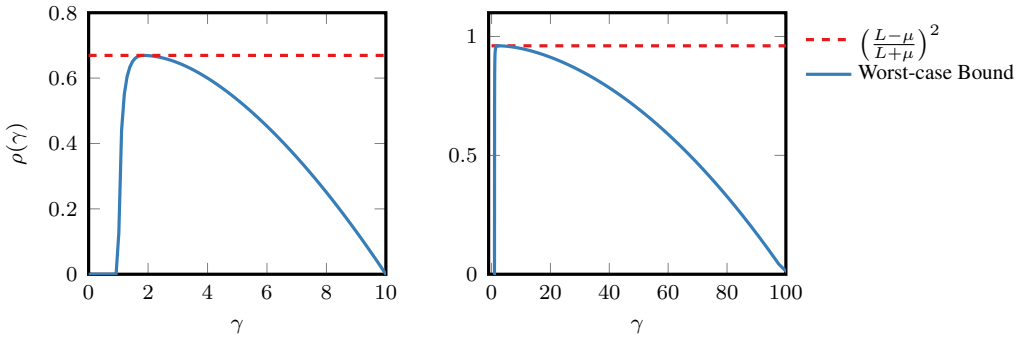


Figure 4.3: We plot $\rho(\gamma)$, by solving an SDP reformulation of (4.18) with $\mu/L = 0.1$ (left) and $\mu/L = 0.01$ (right).

**Remark 4.5.1.** *Note that we directly obtain the tightness of the semidefinite reformulation used to compute $\rho(\gamma)$ numerically (i.e. the numerical solution corresponds to (4.18)), whereas Klerk et al. [2017] had to provide a worst-case example. This also allows obtaining tight reformulations for different criteria (e.g. $\|x_{k+1} - x_*\|^2/\|x_k - x_*\|^2$ with $x_* \in \operatorname{argmin}_x f(x)$ as displayed in Figure 4.4).*

We can look at another criterion such as $\|x_{k+1} - x_*\|^2/\|x_k - x_*\|^2$ with $x_* \in \operatorname{argmin}_x f(x)$ and study the problem

$$\begin{aligned} \rho(\gamma) := \quad & \max \quad \frac{\|x_{k+1} - x_*\|^2}{\|x_k - x_*\|^2} \\ & \text{s.t.} \quad x_{k+1} = x_k - \gamma \nabla f(x_k), \\ & \quad \langle \nabla f(x_{k+1}), \nabla f(x_k) \rangle = 0, \\ & \quad f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d),\ x_k \in \mathbb{R}^d,\ d \in \mathbb{N}. \end{aligned} \tag{4.19}$$

It can be reformulated as a semidefinite problem and Figure 4.4 exhibits numerical solutions for various values of $\gamma$. In particular, we notice that the bounds given by (4.19) are smaller than those obtained using the relaxation of Klerk et al. [2017] (red dashed line).

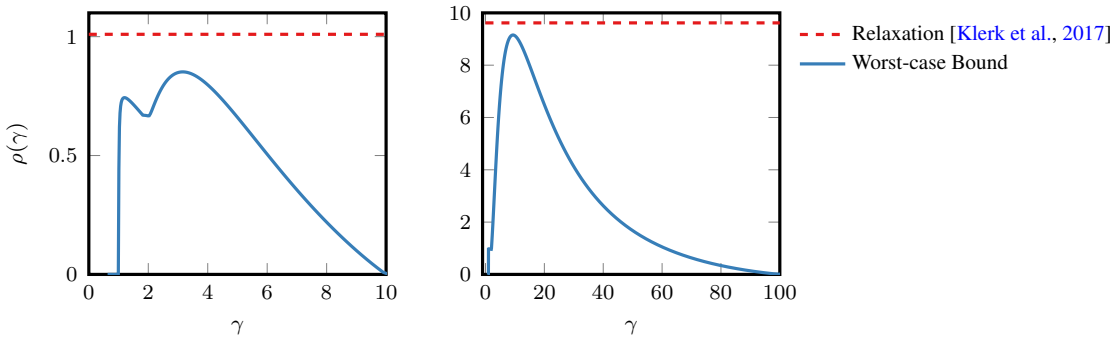


Figure 4.4: We plot $\rho(\gamma)$, by solving an SDP reformulation of (4.19) with $\mu/L = 0.1$ (left) and $\mu/L = 0.01$ (right). The red dashed line corresponds to the bound obtained by solving a relaxed version of the performance estimation problem associated to the exact line search procedure as in Klerk et al. [2017].

### 4.5.2 Conjugate gradient method

In the following we study 2 iterations of a nonlinear conjugate gradient method from Fletcher and Reeves [1964]. Beyond quadratic optimization problems, worst-case guarantees are difficult to obtain in general.

We focus on updates of the form

$$\left\{\begin{array}{rcl} \gamma_k & = & \underset{\gamma}{\operatorname{argmin}} f(x_k - \gamma \nabla f(x_k)) \\ x_{k+1} & = & x_k - \gamma_k \nabla f(x_k) \\ d_{k+1} & = & \nabla f(x_{k+1}) + \beta \nabla f(x_k) \\ \gamma_{k+1} & = & \underset{\alpha}{\operatorname{argmin}} f(x_{k+1} - \alpha d_{k+1}) \\ x_{k+2} & = & x_{k+1} - \gamma_{k+1} d_{k+1}, \end{array}\right. \tag{4.20}$$

where $f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$, $x_k \in \mathbb{R}^d$ and $\beta = \|\nabla f(x_{k+1})\|^2 / \|\nabla f(x_k)\|^2$.

We study worst-case bounds on the ration $(f(x_{k+2}) - f_*)/(f(x_k) - f_*)$ by solving the following problem

$$\begin{array}{ll} \max & \dfrac{f(x_{k+2}) - f_*}{f(x_k) - f_*} \\ \text{s.t.} & \gamma_k = \underset{\gamma}{\operatorname{argmin}} f(x_k - \gamma \nabla f(x_k)), \\ & x_{k+1} = x_k - \gamma_k \nabla f(x_k), \\ & d_{k+1} = \nabla f(x_{k+1}) + \beta \nabla f(x_k), \\ & \gamma_{k+1} = \underset{\gamma}{\operatorname{argmin}} f(x_{k+1} - \gamma d_{k+1}), \\ & x_{k+2} = x_{k+1} - \gamma_{k+1} d_{k+1}, \\ & \|\nabla f(x_{k+1})\|^2 = \beta \|\nabla f(x_k)\|^2, \\ & f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d),\ x_k \in \mathbb{R}^d,\ d \in \mathbb{N}, \end{array} \tag{4.21}$$

for a range of values $\beta$. Note that we could introduce 2 additional parameters to handle the exact line searches as in the previous section, but for simplicity we follow the methodology of Klerk et al. [2017] to handle line searches. We can relax (4.21) as

$$\begin{array}{rll} \rho(\beta) := & \max & \dfrac{f(x_{k+2}) - f_*}{f(x_k) - f_*} \\ & \text{s.t.} & d_{k+1} = \nabla f(x_{k+1}) + \beta \nabla f(x_k), \\ & & \langle \nabla f(x_{k+1}), \nabla f(x_k) \rangle = 0, \\ & & \langle \nabla f(x_{k+1}), x_{k+1} - x_k \rangle = 0, \\ & & \langle \nabla f(x_{k+2}), d_{k+1} \rangle = 0, \\ & & \langle \nabla f(x_{k+2}), x_{k+2} - x_{k+1} \rangle = 0, \\ & & \|\nabla f(x_{k+1})\|^2 = \beta \|\nabla f(x_k)\|^2, \\ & & f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d),\ x_k,\ x_{k+1},\ x_{k+2} \in \mathbb{R}^d,\ d \in \mathbb{N}, \end{array} \tag{4.22}$$

and $\rho(\beta)$ is an upper bound on the optimal value of (4.21). Problem (4.22) can be reformulated as a semidefinite problem (following the steps in Chapter 3 and the references therein). Numerical computations are reported in Figure 4.5. These numerical results tend to show that 2 iterations of conjugate gradient (4.20) yield better worst-case guarantees than 2 iterations of gradient descent with exact line search (red dashed lines in Figure 4.5), with essentially the same cost.

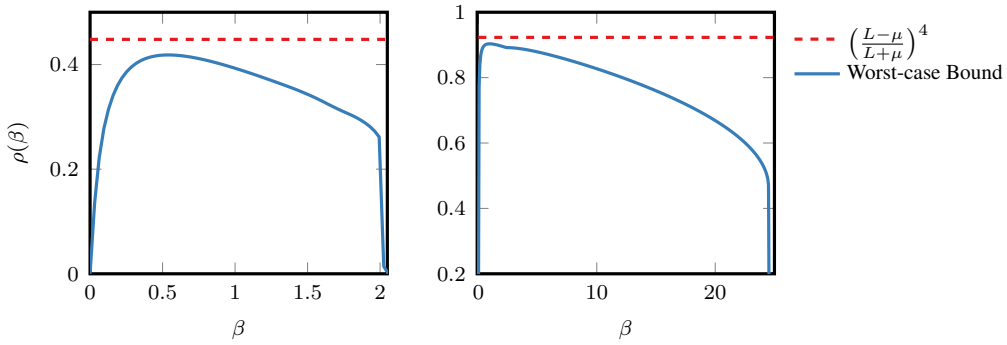


Figure 4.5: We plot $\rho(\beta)$, by solving a SDP reformulation of (4.22) with $\mu/L = 0.1$ (left) and $\mu/L = 0.01$ (right).

### 4.5.3 Regularized Anderson acceleration

Finally, we study worst-case guarantees of a regularized version of the Anderson extrapolation scheme due to Scieur et al. [2020]. We focus an a simple two-step extrapolation procedure (see Chapter 2 for details on Anderson extrapolation) with a regularization proportional to the gradient of the initial iterate, which corresponds to the following updates

$$\left\{ \begin{array}{rcl} x_{k+1} & = & x_k - \frac{1}{L} \nabla f(x_k) \\ c_* & = & \underset{c \in \mathbb{R}}{\operatorname{argmin}} \frac{1}{2} \|(1-c)\nabla f(x_k) + c \nabla f(x_{k+1})\|^2 + \frac{\lambda \|\nabla f(x_k)\|^2}{2} \left((1-c)^2 + c^2\right) \\ x_e & = & (1-c_*) x_k + c_* x_{k+1}, \end{array} \right. \tag{4.23}$$

where $f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$, $x_k \in \mathbb{R}^d$ and $\lambda \geq 0$. Using first-order optimality condition, we can notice that

$$c_* = \underset{c \in \mathbb{R}}{\operatorname{argmin}} \frac{1}{2} \|(1-c)\nabla f(x_k) + c \nabla f(x_{k+1})\|^2 + \frac{\lambda \|\nabla f(x_k)\|^2}{2} \left((1-c)^2 + c^2\right),$$

if and only if

$$\langle \nabla f(x_k), \nabla f(x_k) - \nabla f(x_{k+1}) \rangle + \lambda \|\nabla f(x_k)\|^2 = c_* \left( \|\nabla f(x_k) - \nabla f(x_{k+1})\|^2 + 2\lambda \|\nabla f(x_k)\|^2 \right).$$

We study worst-case bounds on the ratio $\|\nabla f(x_e)\|^2 / \|\nabla f(x_k)\|^2$ (which is often considered when analyzing Anderson acceleration methods, see e.g. Chapter 2) by solving the following problem

$$\begin{aligned} \rho(c) := \quad & \max \quad \frac{\|\nabla f(x_e)\|^2}{\|\nabla f(x_k)\|^2} \\ & \text{s.t.} \quad x_{k+1} = x_k - \tfrac{1}{L} \nabla f(x_k), \\ & \quad\;\; x_e = (1 - c_*) x_k + c_* x_{k+1}, \\ & \quad\;\; \langle \nabla f(x_k), \nabla f(x_k) - \nabla f(x_{k+1}) \rangle + \lambda \|\nabla f(x_k)\|^2 = \\ & \quad\;\; c \left( \|\nabla f(x_k) - \nabla f(x_{k+1})\|^2 + 2\lambda \|\nabla f(x_k)\|^2 \right), \\ & \quad\;\; f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d),\ x_k \in \mathbb{R}^d,\ d \in \mathbb{N}, \end{aligned} \tag{4.24}$$

for various values of $c$. We can reformulate this problem as a semidefinite program (following the steps in Chapter 3 and the references therein), and numerical solutions are displayed in Figure 4.6. We can observe that the worst-case bounds appear very sensitive to the choice of $\lambda$. Indeed, we mainly see two regimes, (i) large $\lambda$'s with little extrapolation (i.e. small $c$) leading to conservative guarantees and (ii) smaller $\lambda$'s leading to larger values of $c$'s and possible explosion of the bounds.

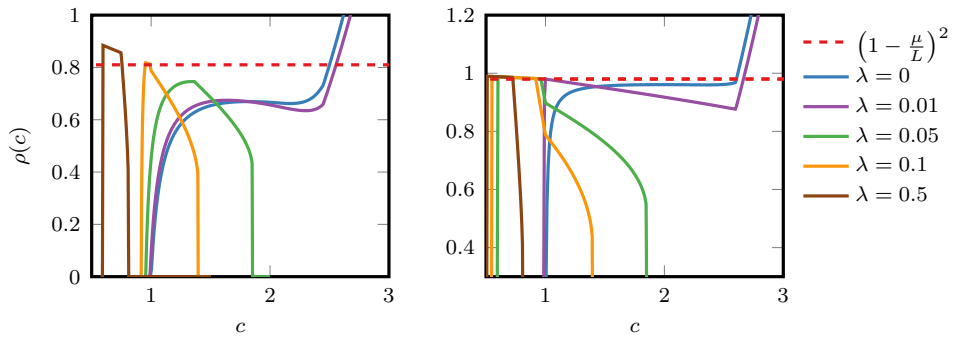


Figure 4.6: We plot $\rho(c)$, by solving a SDP reformulation of (4.24) with $\mu/L = 0.1$ (left) and $\mu/L = 0.01$ (right). The red dashed line correspond to the worst-case bounds on the ratio of gradients for gradient descent.

## 4.6 Numerical experiments

Numerical experiments with our algorithms are provided in Figure 4.7, respectively on least squares, regularized logistic regression and Lasso problems. For solving the Lasso problems, we used a proximal variant of Algorithm 4.3, whose details are provided in Section 4.B. We respectively used the Sonar [Gorman and Sejnowski, 1988] and Musk [Dietterich et al., 1997] datasets.

In the experiments, when no analytical version of $f_*$ was available (for logistic regression and Lasso), we used ad hoc methods to obtain higher precision estimates of $f_*$. As previously discussed, a fundamental next step is to incorporate successive refinements of a lower bound on $f_*$ (a first step in this direction is for example Hazan and Kakade [2019]). One should notice that vanilla Polyak steps without momentum actually perform very well when they apply (see Section 4.C for a discussion on

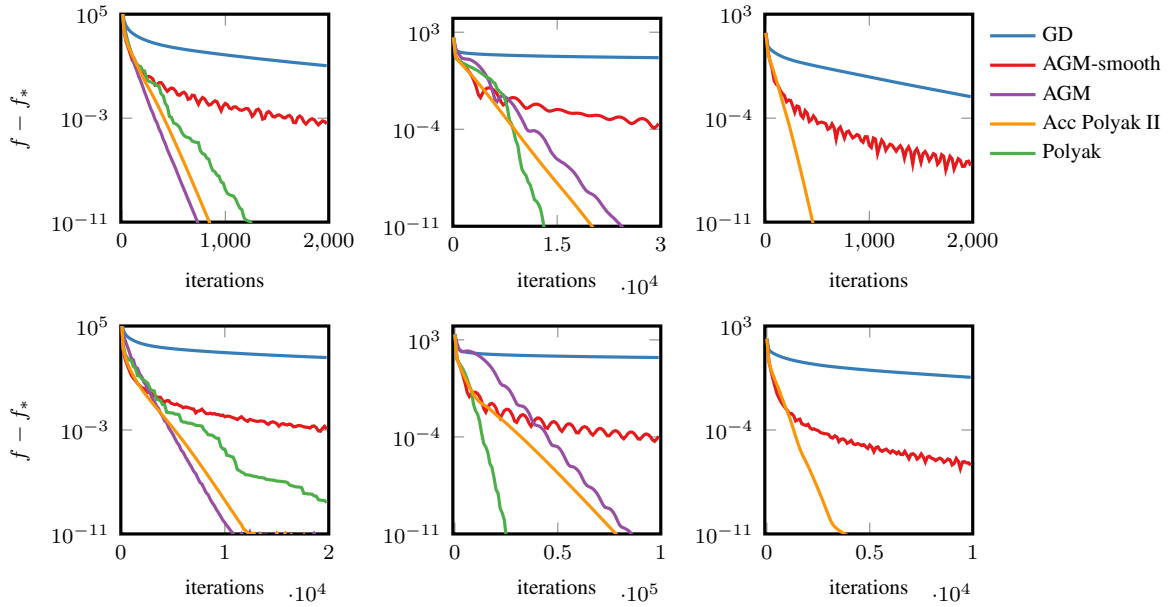


Figure 4.7: Top: Sonar dataset. Bottom: Musk dataset. Left: Least squares. Middle: Logistic regression with Tikhonov regularization (regularization parameter $10^{-3}$). Right: LASSO (regularization parameter 1). For Polyak steps the best iterate is displayed. No tuning in any of the methods.

the performances of vanilla Polyak steps). We believe that modifying the accelerated Polyak so that it also adapts to the Lipschitz constant could make it more competitive, but the current state of the proofs does not allow it yet.

## 4.7 Conclusion and perspectives

We provided a momentum version of the Polyak steps, with an accelerated linear convergence rate. When $f_*$ is available, this method is easy to implement and requires no tuning at all. On the way, we illustrated the methodology that was used for obtaining those rates, for the special case of a gradient method with Polyak steps. This methodology relies on the recent developments on performance estimation problems [Drori and Teboulle, 2014, Taylor et al., 2017c], which we adapted for studying our adaptive methods.

One of the main questions that remains open is to understand whether there exists a way to get the same convergence guarantees without using $f_*$. The robustness result of Lemma 4.3.1 is reassuring in the sense that a misspecified $f_*$ cannot break the algorithm (albeit worsening the convergence rate).

Let us mention that the problem of designing theoretically supported adaptive methods is an open question. We managed to design (Variant II), for which we used our methodology—to find a method that would use Polyak steps to make the primal gap decrease linearly at each iterations—, but designing adaptive accelerated methods appeared as much more daunting task.

Finally, we note that regular Polyak steps do not enjoy a known (working) proximal extension. On the contrary, our results suggest that its accelerated counterparts do work with proximal operators (for minimizing composite objective functions with a non-smooth term). Therefore, developing the theory in this direction is another natural next step.

**Codes** The code used to obtain Figures 4.2, 4.3 and 4.5 to 4.7, Figure 4.C.1 and to verify proofs is available at https://github.com/mathbarre/PerformanceEstimationPolyakSteps/tree/extended.

# Appendices

## 4.A Proof of Lemma 4.3.6

*Proof.* Our statement follows from a weighted sum of inequalities obtained from Lemma 4.1.1:

- smoothness and strong convexity between $y_{k+1}$ and $x_k$, with multiplier $\lambda_1 = 1$:

$$\begin{aligned} f(y_{k+1}) - f(x_k) + \nabla f(y_{k+1})^T(x_k - y_{k+1}) + \tfrac{1}{2L}\|\nabla f(y_{k+1}) - \nabla f(x_k)\|^2 \\ + \tfrac{\mu}{2(1-\frac{\mu}{L})}\|y_{k+1} - x_k - \tfrac{1}{L}(\nabla f(y_{k+1}) - \nabla f(x_k))\|^2 \leq 0, \end{aligned}$$

- smoothness and strong convexity between $x_k$ and $x_*$, with multiplier $\lambda_2 = 1 - \rho$:

$$\begin{aligned} f(x_k) - f_* + \nabla f(x_k)^T(x_* - x_k) + \tfrac{1}{2L}\|\nabla f(x_k)\|^2 \\ + \tfrac{\mu}{2(1-\frac{\mu}{L})}\|x_k - x_* - \tfrac{1}{L}\nabla f(x_k)\|^2 \leq 0, \end{aligned}$$

- convexity between $x_k$ and $y_k$, with multiplier $\lambda_3 = \rho$:

$$f(x_k) - f(y_k) + \nabla f(x_k)^T(y_k - x_k) \leq 0.$$

The weighted sum is a valid inequality given that $\lambda_1, \lambda_2, \lambda_3 \geq 0$:

$$\begin{aligned} 0 \geq \lambda_1 \Big[ & f(y_{k+1}) - f(x_k) + \nabla f(y_{k+1})^T(x_k - y_{k+1}) + \tfrac{1}{2L}\|\nabla f(y_{k+1}) - \nabla f(x_k)\|^2 \\ & + \tfrac{\mu}{2(1-\frac{\mu}{L})}\|y_{k+1} - x_k - \frac{1}{L}(\nabla f(y_{k+1}) - \nabla f(x_k))\|^2 \Big] \\ + \lambda_2 \Big[ & f(x_k) - f_* + \nabla f(x_k)^T(x_* - x_k) + \tfrac{1}{2L}\|\nabla f(x_k)\|^2 \\ & + \tfrac{\mu}{2(1-\frac{\mu}{L})}\|x_k - x_* - \tfrac{1}{L}\nabla f(x_k)\|^2 \Big] \\ + \lambda_3 \Big[ & f(x_k) - f(y_k) + \nabla f(x_k)^T(y_k - x_k) \Big]. \end{aligned}$$

This inequality can be reformulated using the notations

$$\begin{aligned} \Phi(x, y) &= f(y) - f_* + \tfrac{L}{2}\|\tfrac{1}{\sqrt{\rho}}(x - x_*) - \sqrt{\rho}(y - x_*)\|^2 \\ y_{k+1} &= x_k - \tfrac{1}{L}\nabla f(x_k) \\ x_{k+1} &= y_{k+1} + \beta_k(y_{k+1} - y_k) \\ \beta &= \beta_k \end{aligned}$$

in the form

$$\begin{aligned} 0 \geq & \Phi(x_{k+1}, y_{k+1}) - \rho\Phi(x_k, y_k) + \tfrac{1}{2(L-\mu)}\|\nabla f(y_{k+1})\|^2 + \tfrac{1-\rho}{2L}\|\nabla f(x_k)\|^2 \\ & + \tfrac{L(\rho^3 - \beta^2)}{2\rho}\|(y_k - x_*) + \tfrac{\beta\rho - \beta(\beta+1) + \rho^2}{\beta^2 - \rho^3}(x_k - x_*) + \tfrac{\beta^2 - \beta\rho + \beta - \rho^2}{\beta^2 L - L\rho^3}\nabla f(x_k)\|^2 \\ & + \tfrac{L^2(1-\rho)\left(\frac{\mu}{L}\rho(2\beta\rho - \beta(\beta+2) + \rho) + (\rho-1)(\beta-\rho)^2\right)}{2(\rho^3 - \beta^2)(L-\mu)}\|x_k - x_* - \tfrac{1}{L}\nabla f(x_k)\|^2. \end{aligned}$$

It is then direct to reach

$$\begin{aligned}\Phi(x_{k+1},y_{k+1}) \leq & \rho\Phi(x_k,y_k) - \tfrac{1}{2(L-\mu)}\|\nabla f(y_{k+1})\|^2 - \tfrac{1-\rho}{2L}\|\nabla f(x_k)\|^2 \\ & - \tfrac{L\left(\rho^3-\beta^2\right)}{2\rho}\|(y_k-x_*) + \tfrac{\beta\rho-\beta(\beta+1)+\rho^2}{\beta^2-\rho^3}(x_k-x_*) + \tfrac{\beta^2-\beta\rho+\beta-\rho^2}{\beta^2L-L\rho^3}\nabla f(x_k)\|^2 \\ & - \tfrac{L^2(1-\rho)\left(\frac{\mu}{L}\rho(2\beta\rho-\beta(\beta+2)+\rho)+(\rho-1)(\beta-\rho)^2\right)}{2(\rho^3-\beta^2)(L-\mu)}\|x_k-x_*-\tfrac{1}{L}\nabla f(x_k)\|^2 \\ \leq & \rho\Phi(x_k,y_k),\end{aligned}$$

where we used the facts that the following coefficients were nonnegative (proofs below) on the domain of interest:

- $\frac{1}{2(L-\mu)} \geq 0$ (clear from the assumption $\mu \leq L$),
- $\frac{1-\rho}{L} \geq 0$ (clear from $\rho \leq 1$),
- $\frac{L\left(\rho^3-\beta^2\right)}{2\rho} \geq 0$ follows from $\left(\rho^3-\beta^2\right) \geq 0$, proved below,
- $\frac{L^2(1-\rho)\left(\frac{\mu}{L}\rho(2\beta\rho-\beta(\beta+2)+\rho)+(\rho-1)(\beta-\rho)^2\right)}{2(\rho^3-\beta^2)(L-\mu)} \geq 0$ follows from previous points along with

  $$\tfrac{\mu}{L}\rho(2\beta\rho-\beta(\beta+2)+\rho)+(\rho-1)(\beta-\rho)^2 \geq 0,$$

  which is proved below.

The missing proofs are as follow. First, let us define $\kappa := \frac{\mu}{L} \in [0,1]$, the (inverse) condition number, and recall that we want to prove the expressions above to be nonnegative when $\rho = \frac{1}{1+\kappa^{3/4}}$ and $\beta_- \leq \beta \leq \beta_+$ with $\beta_- = \frac{\sqrt{1}-\sqrt[4]{\kappa}}{\sqrt{1}+\sqrt[4]{\kappa}}$ and $\beta_+ = \frac{\sqrt{1}-\sqrt{\kappa}}{\sqrt{1}+\sqrt{\kappa}}$.

- To show that $\rho^3-\beta^2 \geq 0$, let us remark that the expression is a second order polynomial in the variable $\beta$ with negative curvature. Therefore, its minimum values are achieved on the boundary of the interval, and it is sufficient to show $\rho^3-\beta_-^2 \geq 0$ and $\rho^3-\beta_+^2 \geq 0$ for establishing our claim. For the case $\beta=\beta_-$, we get:

  $$\rho^3-\beta_-^2 = \tfrac{\kappa^{1/4}\left(4-8\kappa^{1/4}+9\sqrt{\kappa}-4\kappa^{3/4}-4\kappa+9\kappa^{5/4}-8\kappa^{3/2}+4\kappa^{7/4}-\kappa^2\right)}{\left(1+\kappa^{1/4}\right)^3\left(1-\kappa^{1/4}+\sqrt{\kappa}\right)^3},$$

  and we need to show that $\left(4-8\kappa^{1/4}+9\sqrt{\kappa}-4\kappa^{3/4}-4\kappa+9\kappa^{5/4}-8\kappa^{3/2}+4\kappa^{7/4}-\kappa^2\right)$ is non negative for all $\kappa \in [0,1]$. For showing that, we perform the change of variable $x \leftarrow \kappa^{1/4}$ (which is invertible since $\kappa \in [0,1]$), and study the polynomial

  $$p_1(x) = -x^8+4x^7-8x^6+9x^5-4x^4-4x^3+9x^2-8x+4,$$

  such that

  $$\begin{aligned}p_1(x) &\geq 3x^7-8x^6+9x^5-4x^4-4x^3+9x^2-8x+4 \\ &= 3x^7-8x^6+9x^5-4x^4-4x^3+5x^2+4(1-x)^2 \\ &\geq 3x^7-8x^6+9x^5-4x^4-4x^3+5x^2 \\ &\geq 3x^7-8x^6+9x^5-4x^4+x^3\end{aligned}$$

$$
\begin{aligned}
&= 3x^7 - 8x^6 + 5x^5 + x^3(2x-1)^2 \\
&\geq x^5(3x^2 - 8x + 5) \\
&= x^5(1-x)(5-3x) \\
&\geq 0,
\end{aligned}
$$

hence finally $\rho^3 - \beta_-^2 \geq 0$. For the case $\beta = \beta_+$, we obtain:

$$
\rho^3 - \beta_+^2 = \frac{\sqrt{\kappa}\left(4-3\kappa^{1/4}+6\kappa^{3/4}-3\kappa-3\kappa^{5/4}+6\kappa^{3/2}-\kappa^{7/4}-3\kappa^2+2\kappa^{9/4}-\kappa^{11/4}\right)}{\left(1+\kappa^{1/4}\right)^3\left(1+\sqrt{\kappa}\right)^2\left(1-\kappa^{1/4}+\sqrt{\kappa}\right)^3},
$$

and we need to show that

$$
\left(4 - 3\kappa^{1/4} + 6\kappa^{3/4} - 3\kappa - 3\kappa^{5/4} + 6\kappa^{3/2} - \kappa^{7/4} - 3\kappa^2 + 2\kappa^{9/4} - \kappa^{11/4}\right)
$$

is nonnegative for all $\kappa \in [0,1]$. After changing variable $x \leftarrow \kappa^{1/4}$ (which is invertible since $\kappa \in [0,1]$), we study the polynomial

$$
p_2(x) = -x^{11} + 2x^9 - 3x^8 - x^7 + 6x^6 - 3x^5 - 3x^4 + 6x^3 - 3x + 4
$$

such that

$$
\begin{aligned}
p_2(x) &\geq x^9 - 3x^8 - x^7 + 6x^6 - 3x^5 - 3x^4 + 6x^3 - 3x + 4 \\
&\geq x^9 - 3x^8 - x^7 + 6x^6 - 3x^5 - 3x^4 + 6x^3 + 1 \\
&\geq x^9 - 3x^8 - x^7 + 6x^6 + 1 \\
&\geq x^9 + 2x^6 + 1 \\
&\geq 0,
\end{aligned}
$$

hence $\rho^3 - \beta_+^2 \geq 0$.

- Similarly, the expression $p_3(\kappa) = \left(\kappa\rho(2\beta\rho - \beta(\beta+2) + \rho) + (\rho-1)(\beta-\rho)^2\right)$ is also a second order polynomial in $\beta$, with leading coefficient

$$
-(1-\rho) - \kappa\rho \leq -(1-\rho) \leq 0.
$$

Therefore, this quadratic function is also concave and we only need to verify the inequality on the boundary of the interval $[\beta_-, \beta_+]$. In the case $\beta = \beta_-$, we get:

$$
p_3(\beta_-) = \frac{\left(1-\sqrt{\kappa}+\kappa^{3/4}\right)\kappa^{7/4}}{\left(1+\kappa^{1/4}\right)^3\left(1-\kappa^{1/4}+\sqrt{\kappa}\right)^3} \geq 0.
$$

For case $\beta = \beta_+$, we obtain:

$$
p_3(\beta_+) = \frac{\kappa^{3/2}\left(4-7\kappa^{1/4}+4\sqrt{\kappa}+5\kappa^{3/4}-7\kappa+3\kappa^{5/4}+2\kappa^{3/2}-\kappa^{7/4}+\kappa^2\right)}{\left(1+\kappa^{1/4}\right)^3\left(1+\sqrt{\kappa}\right)^2\left(1-\kappa^{1/4}+\sqrt{\kappa}\right)^3},
$$

and we need to show that $\left(\kappa^2 - \kappa^{7/4} + 2\kappa^{3/2} + 3\kappa^{5/4} - 7\kappa + 5\kappa^{3/4} + 4\sqrt{\kappa} - 7\sqrt[4]{\kappa} + 4\right)$ is nonnegative for $\kappa \in [0,1]$. We change variables $x \leftarrow \kappa^{1/4}$ (which is invertible since $\kappa \in [0,1]$), and study the polynomial

$$
p_4(x) = x^8 - x^7 + 2x^6 + 3x^5 - 7x^4 + 5x^3 + 4x^2 - 7x + 4
$$

on the interval $[0, 1]$:

$$\begin{aligned} p_4(x) &= x^8 - x^7 + 2x^6 + 3x^5 - 7x^4 + 5x^3 + x + 4(1-x)^2 \\ &\geq x^3(x^5 - x^4 + 2x^3 + 3x^2 - 7x + 5) \\ &= x^3(x^5 - x^4 + 2x^3 - x^2 + x + 1 + 4(1-x)^2) \\ &\geq x^3(x^5 + x^3 + 1 + 4(1-x)^2) \\ &\geq 0, \end{aligned}$$

hence $p_3(\beta_+) \geq 0$, which concludes the proof.

■

## 4.B Proximal variants

A natural extension of smooth and strongly convex optimization is the case composite optimization

$$\min_{x \in \mathbb{R}^d} \{F(x) \equiv f(x) + g(x)\},$$

where $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ and $g \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ is a proper convex function with proximal operator available.

**Proximal accelerated gradient method** (Algorithm 4.B)

**Input:**

- Objective function: $f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$, $g \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$.
- Initial guess: $x_0 \in \mathbb{R}^d$.
- Optimal objective value: $f_* \in \mathbb{R}$.
- Smoothness parameter: $L \in \mathbb{R}^*$.

**Initialization:**

$$y_0 = x_0$$

**Run:**

For $k = 0, \ldots$ :

$$y_{k+1} = \operatorname{prox}_{\frac{g}{L}}(x_k - \tfrac{1}{L}\nabla f(x_k))$$

Compute $\tilde{\mu}_k$ and $\beta_k = \frac{\sqrt{L} - \sqrt{\tilde{\mu}_k}}{\sqrt{L} + \sqrt{\tilde{\mu}_k}}$

$$x_{k+1} = y_{k+1} + \beta_k(y_{k+1} - y_k)$$

End For

**Output:** $y_{k+1}$

We used the proximal version of AGM with constant momentum. It is of the same form as Algorithm 4.3 but the gradient step is combined with a proximal step. We extended our estimate $\tilde{\mu}_k$ the following way. Given $F \equiv f + g$ where $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ and $g$ a proper convex function that is proximable,

$$\tilde{\mu}_k = \frac{\mathcal{D}(y_{k+1}, L)}{2(F(y_{k+1}) - F_*)},$$

where

$$\mathcal{D}(x, L) = -2L \min_y \left[\langle \nabla f(x), y - x\rangle + \tfrac{L}{2}\|x - y\|^2 + g(y) - g(x)\right].$$

Notice that when $g = 0$ the previous formula is exactly (Acc. Variant I). Also, when they are well defined these estimates still belong to $[\mu, L]$ [Karimi et al., 2016].

## 4.C Study of standard Polyak steps

In this section, we briefly study the standard choice of step sizes (Polyak).

### 4.C.1 Practical behavior

From numerical experiments, we noticed that (Variant I) was actually typically performing only slightly better than vanilla gradient descent. From a worst-case point of view, this is expected. However, our experiments (see Figures 4.1 and 4.7) suggest that regular Polyak steps (Polyak) actually perform much better than one could expect from its worst-case guarantees.

In this section, we provide a tentative explanation of this behavior, through experiments on a toy example. Figure 4.C.1 (top) was obtained by running the methods on a least squares problem (we used a rescaled version of the Sonar dataset, with regularity parameters $L = 1$ and $\mu = 0.01$).

Similar in spirit as in Figure 4.2 (left), we provide, in Figure 4.C.1, the worst-case ratio of $\|x_{k+1} - x_*\|^2 / \|x_k - x_*\|^2$ (by solving (4.12) numerically for regular Polyak steps). One can observe that the worst case rate (using distances to optimum as the criterion) is slightly worse than that of (Variant I) (note that this rate can be improved through the use of refined Lyapunov functions).

In Figure 4.C.1, we provide the distributions of step size magnitudes observed through the optimization process on the toy example. One can notice that the distribution does not fully concentrate around the worst-case value (the value of $\gamma$ that achieves the worst-case) for (Polyak). A large proportion of effective step size values are even located in regions of fast convergence. On the contrary, for (Variant I), the distribution is much more concentrated around its worst-case. Those distributions strongly suggest that worst case analyses might not be the best way to explain the good practical behaviors of such adaptive methods.

We show in the next section that Polyak steps methods with step sizes (Polyak) and (Variant I) behave similarly in the worst-case, although their practical performances appear quite different.

### 4.C.2 A worst-case example

Note that the numerical worst-case guarantee for standard Polyak steps in Figure 4.C.1 is only valid for 1 iteration of the method. The worst-case behavior after $N$ iterations could potentially be much better. However, as for Polyak steps with (Variant I), we can exhibits a function on which the convergence rate for the distance to the optimum after $N$ iterations is $\frac{(L-\mu)^{2N}}{(L+\mu)^{2N}}$.

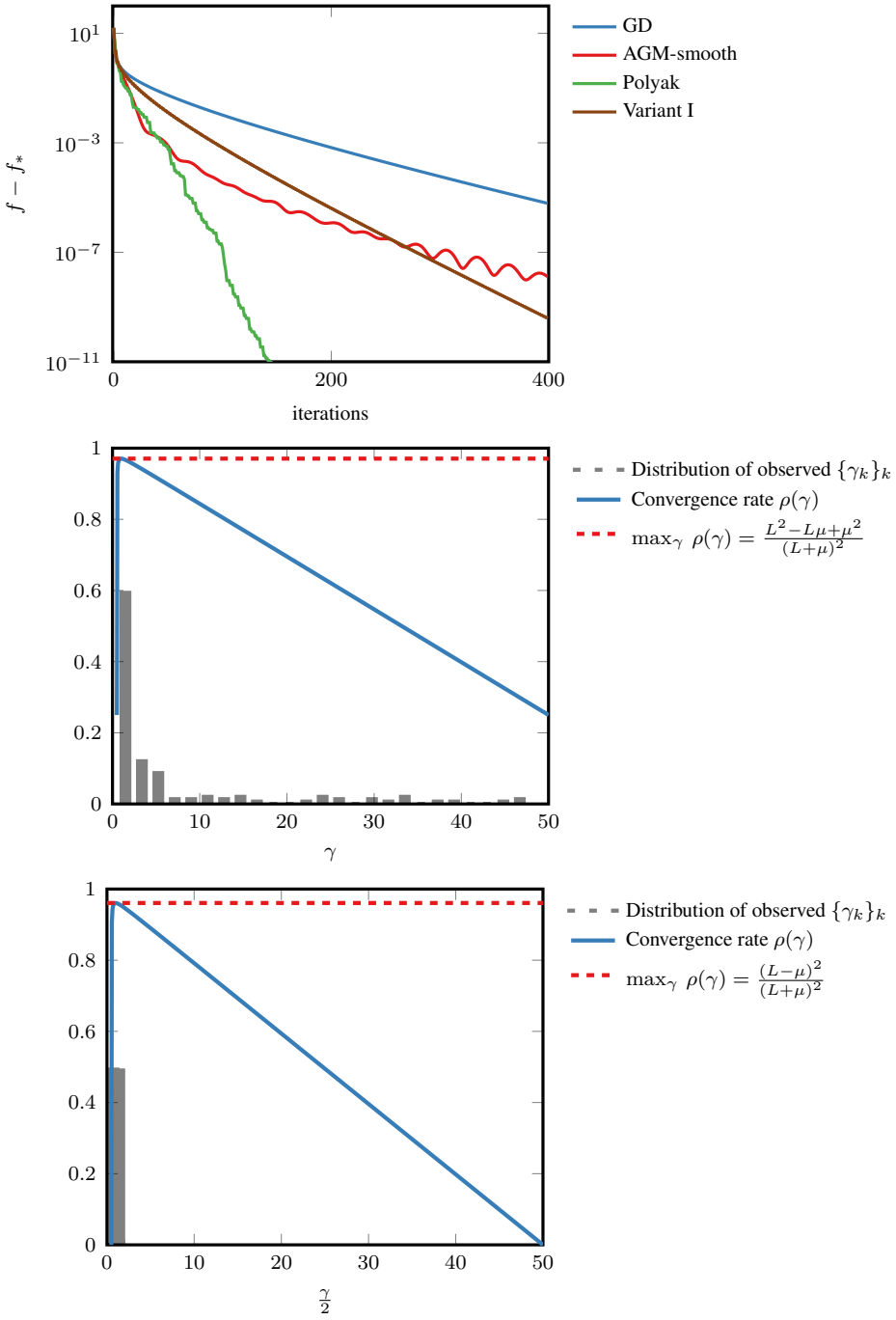


Figure 4.C.1: Top: Least squares on rescaled Sonar dataset ($L = 1$ and $\mu = 0.01$). Middle: $\rho(\gamma)$ for (Polyak) (blue)—computed numerically following the methodology of Section 4.4 with fixed $L = 1$ and $\mu = 0.01$. Distribution of effective step size magnitudes (black) used throughout the 150 iterations of (Polyak) appearing in (top). Bottom: $\rho(\gamma)$ for (Variant I) (blue)—with $L = 1$ and $\mu = 0.01$. Distribution of effective step size magnitudes (black) used throughout the 400 iterations of (Variant I) appearing in (top).

**Proposition 4.C.1.** *Let $d \geq 2$, $N \in \mathbb{N}$, $x_0 \in \mathbb{R}^d$, $0 < \mu \leq \frac{L}{3}$, there exists $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$ such that, the output $x_N$ of Algorithm 4.2 with step sizes (Polyak) applied to $f$ satisfies*

$$\|x_N - x_*\|^2 = \left(\tfrac{(L-\mu)^2}{(L+\mu)^2}\right)^N \|x_0 - x_*\|^2,$$

*where $x_* = \operatorname{argmin}_x f(x)$.*

*Proof.* The structure of the worst-case example that we present here is similar to that in Section 4.2.1. Indeed, we consider the 2-dimensional quadratic function

$$f(x) = \tfrac{1}{2} x^T H x,$$

with

$$H = \tfrac{1}{3(L+\mu)} \begin{pmatrix} 4L\mu & -\sqrt{L\mu(L-3\mu)(3L-\mu)} \\ -\sqrt{L\mu(L-3\mu)(3L-\mu)} & 4L^2 - (L-3\mu)(L+\mu) \end{pmatrix}.$$

We can compute $\operatorname{Tr}(H) = L + \mu$ and $\det(H) = L\mu$, therefore $H$ has $L$ and $\mu$ as eigenvalues and $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$. Let $x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, and $\{x_k\}_k$ define as

$$x_{k+1} = x_k - \tfrac{f(x_k) - f_*}{\|\nabla f(x_k)\|^2} \nabla f(x_k).$$

As in the proof of Proposition 4.2.1, we can show that the Polyak steps keep the value $\frac{2}{L+\mu}$, that $x_2 = \frac{(L-\mu)^2}{(L+\mu)^2} x_0$ and $\|x_1 - x_*\|^2 = \frac{(L-\mu)^2}{(L+\mu)^2} \|x_0 - x_*\|^2$, which leads to

$$\begin{aligned} x_{2k+1} &= \tfrac{(L-\mu)^2}{(L+\mu)^2}^{2k} x_1 \\ x_{2k+2} &= \tfrac{(L-\mu)^2}{(L+\mu)^2}^{2k+2} x_0, \end{aligned} \tag{4.25}$$

and the proof is concluded similarly to the proof of Proposition 4.2.1.

■

# Chapter 5

# Principled Analyses of First-Order Methods with Inexact Proximal Operations

*Proximal* operations are among the most common primitives appearing in both practical and theoretical (or high-level) optimization methods. This basic operation typically consists in solving an intermediary (hopefully simpler) optimization problem. In this chapter, we survey notions of inaccuracies that can be used when solving those intermediary optimization problems. Then, we show that worst-case guarantees for algorithms relying on such inexact proximal operations can be systematically obtained through a generic procedure based on semidefinite programming. This methodology is primarily based on the approach introduced by Drori and Teboulle Drori and Teboulle [2014] and on convex interpolation results [Taylor et al., 2017c], and allows producing non-improvable worst-case analyses. In other words, for a given algorithm, the methodology generates both worst-case certificates (i.e., proofs) and problem instances on which those bounds are achieved.

Relying on this methodology, we study numerical worst-case performances of a few basic methods relying on inexact proximal operations including accelerated variants, and design a variant with optimized worst-case behavior. We further illustrate how to extend the approach to support strongly convex objectives by studying a simple relatively inexact proximal minimization method.

**Contributions** We can summarize our main contributions as follows:

(i) We review the literature on inexactness criteria in proximal computations and show that many of them can be expressed in a generic way.

(ii) We show that the performance estimation approach [Drori and Teboulle, 2014, Taylor et al., 2017c] can be used for analyzing algorithms with inexact proximal computations.

(iii) We provide an accelerated relatively inexact proximal point method with optimized worst-case guarantees.

**Organization** This work is organized as follows: Section 5.1 introduces the notions of inexact proximal computations and computer-assisted worst-case analyses. In Section 5.2 we survey common and natural notions of inaccuracies for proximal operations. Then, because of the structure of the inexactness criteria, we show in Section 5.3 that worst-case analyses of algorithms relying on such inexact

proximal operations can be studied with performance estimation, which we later illustrate through several examples. Finally, we use the approach to optimize the parameters of a method relying on inexact proximal operations, in Section 5.4. Strongly convex objectives are treated in Section 5.5, before drawing some conclusions in Section 5.6.

## 5.1 Introduction

Proximal operations serve as base primitives in many conceptual and practical optimization methods. Formally, given a closed, proper, convex function $h : \mathbb{R}^d \to \mathbb{R}$, the proximal map of $h$, denoted by $\text{prox}_{\lambda h} : \mathbb{R}^d \to \mathbb{R}^d$, is

$$\text{prox}_{\lambda h}(z) = \underset{x \in \mathbb{R}^d}{\text{argmin}} \left\{ \lambda h(x) + \tfrac{1}{2}\|x - z\|^2 \right\},$$

where $\lambda$ is a step size. In ideal situations, proximal operations are accessed through analytical expressions (see e.g., Chierchia et al. [2020]). However, in many cases, proximal steps have to be computed only approximately (e.g., via iterative methods). Although those problems may often be solved efficiently, one has to take those inaccuracies into account while analyzing the corresponding algorithms, in order to design methods that are sufficiently robust, and for avoiding solving the proximal subproblem to an unnecessary high precision. Those topics are motivated in different areas of the optimization literature, in particular for augmented Lagrangian techniques (e.g., when the augmented Lagrangian has to be solved numerically), and in the context of splitting methods when proximal operators are complicated, or expensive, to compute.

In this chapter, we show that the performance estimation framework, originating from Drori and Teboulle [2014], can be used for studying algorithms whose base operations are approximate proximal operators. We illustrate the approach by studying numerical worst-case guarantees on various methods from the literature, and by designing an optimized inexact proximal minimization method. On the way, we survey notions of approximate proximal operators that are used in the literature.

### 5.1.1 Motivations and contributions

The main motivation of this work is to improve our capabilities of performing worst-case analyses of algorithms involving inexact proximal operations. Relying on the idea of performance estimation, and convex interpolation, we show that such analyses (i) can be completed in a principled way, and (ii) that semidefinite programming can help in the process of designing the proof. We first illustrate the approach on a variant of the inexact proximal point algorithm under a simple model of inaccuracy, and further explore the worst-case behavior of a few accelerated inexact proximal methods from Salzo and Villa [2012], Monteiro and Svaiter [2013]. Then, we use it for designing an optimized relatively inexact method under a generic primal-dual inaccuracy model. Finally, we use a simple inexact proximal minimization method for showing how to extend the methodology to treat strongly convex objectives.

### 5.1.2 Relationships with previous works

Proximal operations, originally introduced by Moreau [Moreau, 1962, 1965], serve as base primitives in many conceptual and practical algorithms. In optimization, its use is omnipresent and originally attributed to Martinet [Martinet, 1970, 1972] and Rockafellar [Rockafellar, 1976a,b]. Successful examples of algorithms relying on proximal operators include proximal gradient methods [Bruck Jr, 1975,

Lions and Mercier, 1979, Passty, 1979, Beck and Teboulle, 2009, Nesterov, 2013], the celebrated alternating direction method of multipliers [Fortin and Glowinski, 1983, Gabay, 1983], the related Douglas-Rachford splitting [Douglas and Rachford, 1956, Lions and Mercier, 1979, Eckstein and Bertsekas, 1992], and many other *splitting methods* [Lions and Mercier, 1979, Eckstein, 1989]. This type of methods are abundantly used in the optimization literature, and lies at the heart of many optimization paradigms that includes distributed/decentralized optimization (e.g., through operator splitting), augmented Lagrangian techniques [Rockafellar, 1973, 1976a, Iusem, 1999, Eckstein and Silva, 2013], and other meta-algorithms, such as "Catalyst" [Lin et al., 2015, 2018]. The many aspects of their theoretical and practical uses are heavily covered in the literature, and we defer those discussions to surveys on such topics [Boyd et al., 2011, Combettes and Pesquet, 2011, Eckstein and Yao, 2012, Parikh and Boyd, 2014, Ryu and Boyd, 2016] and the references therein.

**Proximal operations and inexactness** Using inexact solutions to proximal operations is not a new idea. First analyses of approximate proximal algorithms for monotone inclusions and optimization problems emerged in Rockafellar [1976b], and this topic appeared in many works since then (see e.g., Güler [1992], Salzo and Villa [2012], Auslender [1987], Solodov and Svaiter [2001], Fuentes et al. [2012], Correa and Lemaréchal [1993], Solodov and Svaiter [2000b,a]). Many notions of inaccuracies are also already covered in the literature. In particular, those notions were applied to the proximal point algorithm Burachik et al. [1997], Eckstein [1998], Solodov and Svaiter [1999a], Monteiro and Svaiter [2013], inexact splitting scheme such as forward-backward splitting (and its accelerated variants) [Schmidt et al., 2011, Villa et al., 2013, Millán and Machado, 2019, Bello-Cruz et al., 2020], Douglas-Rachford [Eckstein and Yao, 2017, 2018, Svaiter, 2018, Alves et al., 2019], three-operator splitting [Zong et al., 2018], online optimization [Dixit et al., 2019, Ajalloeian et al., 2020, Bastianello et al., 2020], and for designing meta-algorithms such as the hybrid approximate extragradient method [Solodov and Svaiter, 1999a, Monteiro and Svaiter, 2010, 2013, Alves and Marcavillaca, 2019], and "Catalyst" [Lin et al., 2015, 2018]. Inexact proximal operations are also closely related to the theory of $\varepsilon$-subdifferentials, introduced in Brøndsted and Rockafellar [1965], and to their inexact gradient and subgradient methods (see e.g., Simonetto and Jamali-Rad [2016], Millán and Machado [2019], Devolder [2013], Devolder et al. [2014]). Finally, let us mention higher-order proximal methods, that are introduced in Nesterov [2020b,a], and also used together with notions for approximating them.

**Monotone inclusions** Inexact proximal methods were also studied in many works in the context of monotone operators and monotone inclusion problems [Rockafellar, 1976b] (see e.g., Bauschke and Combettes [2011] for the general topic of monotone operators, or the nice tutorial Ryu and Boyd [2016]). This was often done through notions of enlargements [Burachik et al., 1998, 1997, 2015], see for example Solodov and Svaiter [1999a, 2001], Burachik et al. [1999], Alves and Marcavillaca [2019], Monteiro and Svaiter [2010], Boţ and Csetnek [2015]. Though we are not going to work with monotone operators and inclusions, there is no apparent obstacle in applying the methodology presented here directly for dealing with inexactness in such setups.

**Computer-assisted analyses** Using semidefinite programming for obtaining worst-case guarantees in the context of first-order optimization schemes dates back to Drori and Teboulle [2014], via so-called *performance estimation problems* (PEPs), which they use to provide novel analyses of gradient, heavy-ball and accelerated gradient methods (see Polyak [1964], Nesterov [1983]). Performance estimation problems were coupled with "convex interpolation" results in Taylor et al. [2017c,a], allowing the PEP

approach to be guaranteed to generate *tight* worst-case certificates. For obtaining simpler proofs, performance estimation problems can be used for designing potential functions Taylor and Bach [2019]. This idea is closely related to that based on *integral quadratic constraints* (IQCs), originally coined in control theory [Megretski and Rantzer, 1997], and which were introduced for analyzing linearly-converging first-order methods in Lessard et al. [2016]; and later extended to deal with sublinear convergence rates Hu and Lessard [2017]. We will not further discuss IQCs here, as the current framework essentially relies on PEPs. Those methodologies being closely related, the developments below could be formulated, instead, in control-theoretic terms.

Let us mention that the PEP methodology was already taken further in different directions, as for example in the context of monotone inclusions: for the three operator splitting [Ryu et al., 2020], proximal point algorithm [Gu and Yang, 2019a,b], and accelerated variants [Kim, 2021]. The methodology was also used in a saddle-point setting in [Drori, 2014, Section 4.6] and for studying worst-case properties of fixed point iterations [Lieder, 2020]. Both IQCs and PEPs were also already used for performing algorithmic design in different settings, starting through the works by Drori and Teboulle [2014], Kim and Fessler [2016], Drori and Teboulle [2016] and taken further in different directions [Taylor et al., 2017a, Kim and Fessler, 2018, Van Scoy et al., 2018, Drori and Taylor, 2020, Kim and Fessler, 2021, Kim, 2021, Ryu and Vũ, 2019]. The methodology was also used in the context of multiplicative gradient noise [Klerk et al., 2020, 2017, Cyrus et al., 2018], and Bregman gradient methods [Dragomir et al., 2021].

### 5.1.3 Preliminary material

We denote by $\mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$ the set of closed proper $\mu$-strongly convex functions on $\mathbb{R}^d$ with $0 \leq \mu < \infty$, and by $\mathcal{F}_{0,\infty}(\mathbb{R}^d)$ the corresponding subset of closed, proper and convex functions. Depending on the context, we will also use the notation $\partial h(x)$ for denoting the subdifferential of $h$ at $x$, or for abusively denoting a particular subgradient of $h$ at $x$, for notational convenience. For $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$, the proximal problem can be formulated through a primal, a saddle point, or a dual formulation, as follows:

$$\min_x \{\Phi_p(x;z) \equiv \lambda h(x) + \tfrac{1}{2}\|x-z\|^2\} \tag{P}$$

$$\max_v \min_x \{\Phi(x,v;z) \equiv \lambda h(x) + \langle \lambda v; z-x\rangle - \tfrac{1}{2}\|\lambda v\|^2\} \tag{SP}$$

$$\max_v \{\Phi_d(v;z) \equiv -\lambda h^*(v) - \tfrac{1}{2}\|\lambda v - z\|^2 + \tfrac{1}{2}\|z\|^2\}, \tag{D}$$

where $h^* \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ denotes the Fenchel conjugate of $h$. In this setting, a sufficient condition for having no duality gap is that $\mathrm{ri}(\mathbf{dom}\, h) \neq \emptyset$ (see e.g., Rockafellar [1996, Corollary 31.2.1], or discussions in Chambolle and Pock [2016, Section 3.5]). In the following sections, we examine natural approximate optimality conditions for those three problems. Let us recall a few relations between their optimal solutions. First, first-order optimality conditions along with Fenchel conjugation allows writing

$$x = \mathrm{prox}_{\lambda h}(z) \Leftrightarrow \tfrac{z-x}{\lambda} \in \partial h(x) \Leftrightarrow x \in \partial h^*(\tfrac{z-x}{\lambda}) \Leftrightarrow \tfrac{z-x}{\lambda} = \mathrm{prox}_{h^*/\lambda}(\tfrac{z}{\lambda}).$$

By noting the last equality can be written as $\frac{z-\mathrm{prox}_{\lambda h}(x)}{\lambda} = \mathrm{prox}_{h^*/\lambda}(\frac{z}{\lambda})$, we arrive to Moreau's identity

$$\mathrm{prox}_{\lambda h}(z) + \lambda \mathrm{prox}_{h^*/\lambda}(\tfrac{z}{\lambda}) = z \tag{Moreau}$$

and to the corresponding identity in terms of function values:

$$h(\mathrm{prox}_{\lambda h}(z)) + h^*(\mathrm{prox}_{h^*/\lambda}(\tfrac{z}{\lambda})) = \langle \mathrm{prox}_{\lambda h}(z); \mathrm{prox}_{h^*/\lambda}(\tfrac{z}{\lambda})\rangle.$$

Though not being mandatory for the understanding of the material covered in the sequel, a great deal of simplifications in the exposition (particularly in the algorithmic analyses) can be obtained through the notion of $\varepsilon$-subdifferentials [Brøndsted and Rockafellar, 1965].

**Definition 5.1.1** (Section 3 of Brøndsted and Rockafellar [1965]). *Let $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$. For any $\varepsilon \geq 0$, we denote by $\partial_\varepsilon h(x)$ the $\varepsilon$-subdifferential of $h$ at $x \in \mathbb{R}^d$:*

$$\begin{aligned}\partial_\varepsilon h(x) &= \{g \,|\, h(z) \geq h(x) + \langle g; z - x\rangle - \varepsilon \quad \forall z \in \mathbb{R}^d\} \\ &= \{g \,|\, h(x) + h^*(g) - \langle g; x\rangle \leq \varepsilon\}.\end{aligned}$$

*Any $g \in \partial_\varepsilon h(x)$ is called an $\varepsilon$-subgradient of $h$ at $x \in \mathbb{R}^d$.*

Before finishing this section, let us note that the symmetry of the second equality in the definition implies $g \in \partial_\varepsilon h(x) \Leftrightarrow x \in \partial_\varepsilon h^*(g)$.

## 5.2 Notions of inexactness for proximal operators

Our main motivation in this section is to survey the main natural notions of inexact proximal operations that can be used in practical applications. In particular, when solving a proximal subproblem through an iterative method, we want to be able to assess the quality of an approximate solution. Therefore, it is important to have accuracy requirements that can be evaluated in practice, and which do not depend on quantities to are generally unknown to the user, such as the exact solution to the proximal subproblem, or an optimal function value. A natural way to design such candidates accuracy conditions is to inspect optimality conditions of the proximal subproblem, and to require our approximate solutions to the subproblems to satisfy them within an appropriate accuracy. We focus on the optimization settings, but many notions extend to the monotone operator world either directly or using concepts of enlargements [Burachik et al., 1997, 1998].

Before proceeding, note that all notions do not have the same practical implications, as some might for example require having access to the dual problem (D), or having access to $h^*$, whereas other do not. In addition, it might be easy to find approximate solutions for certain accuracy requirements, but hard to find candidates for others, depending on the target application.

In this section, we propose a list of natural notions for measuring inaccuracies within proximal operations. Those notions are not new, and our intent here is to list them in a systematic way, and to show (in the next section) that worst-case analyses of natural algorithms relying on such notions can be studied by following the same *principled* steps.

Our starting point is to express optimality conditions for the proximal subproblem in its different forms (P), (SP), and (D), as follows.

- First-order optimality conditions of (SP) can be written as

$$\begin{cases} x = \operatorname{prox}_{\lambda h}(z) \\ v = \operatorname{prox}_{h^*/\lambda}(\frac{z}{\lambda}) \end{cases} \Leftrightarrow 0 \in \begin{pmatrix} \partial_x \Phi(x, v; z) \\ \partial_v(-\Phi(x, v; z)) \end{pmatrix},$$

  which can equivalently be formulated as the optimality conditions of either (P) or (D):

$$\begin{cases} 0 = \|w - v\| \text{ for some } w \in \partial h(x) \Leftrightarrow 0 = \|u - x\| \text{ for some } u \in \partial h^*(v), \\ 0 = \|x - z + \lambda v\|. \end{cases}$$

- Assuming no duality gap occurs between (P) and (D) (see Section 5.1.3), one can write the zeroth-order optimality conditions (i.e., the primal-dual gap) for (SP)

$$\begin{cases} x = \mathrm{prox}_{\lambda h}(z) \\ v = \mathrm{prox}_{h^*/\lambda}(\frac{z}{\lambda}) \end{cases} \Leftrightarrow \Phi_p(x; z) - \Phi_d(v; z) = 0,$$

which can explicitly be written as

$$\Phi_p(x; z) - \Phi_d(v; z) = \lambda h(x) + \lambda h^*(v) - \lambda\langle x; v\rangle + \tfrac{1}{2}\|x - z + \lambda v\|^2.$$

We observe in the previous primal-dual gap expression that it decomposes as the sum of two nonnegative quantities $\lambda h(x) + \lambda h^*(v) - \lambda\langle x; v\rangle$ and $\frac{1}{2}\|x - z + \lambda v\|^2$. In particular, the first term controls how far is $v$ from $\partial h(x)$. Indeed, first-order optimality conditions applied to the definition of the Fenchel-Legendre transform (see e.g., Rockafellar [1996, Theorem 23.5]) gives

$$0 = \lambda h(x) + \lambda h^*(v) - \lambda\langle x; v\rangle \Leftrightarrow v \in \partial h(x) \Leftrightarrow x \in \partial h^*(v).$$

Moreover, when this term is nonzero, one can express the relationship between $x$ and $v$ through $\varepsilon$-subdifferentials (see Definition 5.1.1) as

$$h(x) + h^*(v) - \langle v; x\rangle \leq \varepsilon \Leftrightarrow v \in \partial_\varepsilon h(x) \Leftrightarrow x \in \partial_\varepsilon h^*(v).$$

In other word, for any primal-dual pair $(x, v)$, $v$ is always an $\varepsilon$-subgradient of $h$ at $x$ with $\varepsilon = h(x) + h^*(v) - \langle x; v\rangle$ (which is finite when $v \in \mathbf{dom}\, h^*$).

Those elements motivate measuring inaccuracies simultaneously in two ways:

(i) requiring $\|x - z + \lambda v\|$ being small enough (i.e., requiring (Moreau) to hold approximately), and

(ii) requiring either $v$ being close enough to $\partial h(x)$, and/or how $x$ being close enough to $\partial h^*(v)$. Via the primal-dual gap formulation, this is done by requiring $h(x) + h^*(v) - \langle v; x\rangle$ to be small enough. In first-order optimality conditions, this could be done by requiring $\|v - w\|$ to be small enough for some $w \in \partial h(x)$ or $\|x - u\|$ to be small enough for some $u \in \partial h^*(y)$.

Note that when either the candidate dual solution satisfies $v \in \partial h(x)$, or the candidate primal solution satisfies $x \in \partial h^*(v)$ (for example if the proximal subproblem is solved via a purely primal, or purely dual, method), then the only term that needs to be controlled is that of (i). In the case where either the approximate dual solution is chosen as $v = \frac{z-x}{\lambda}$ or the approximate primal solution is chosen as $x = z - \lambda v$, the only term that needs to be controlled is (ii), as (i) is automatically 0. In other cases, both terms need to be controlled.

### 5.2.1 A few observable notions of inexactness

In this section, we are interested in inexactness notions that do no require knowledge on $\mathrm{prox}_{\lambda h}(z)$ or $\mathrm{prox}_{h^*/\lambda}(\frac{z}{\lambda})$ to be evaluated. In what follows, we denote the primal-dual gap by

$$\mathrm{PD}_{\lambda h}(x, v; z) := \Phi_p(x; z) - \Phi_d(v; z),$$

and the Moreau gap by

$$\mathrm{M}_\lambda(x, v; z) := \|x - z + \lambda v\|^2,$$

for convenience, and we recall a property on the primal-dual gap that was stated earlier in Section 5.2 but that is key to compare it with $\varepsilon$-subgradient based criterion in the literature.

**Lemma 5.2.1.** *Let $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$, $\varepsilon \geq 0$, $x, v, z \in \mathbb{R}^d$. If $v \in \partial_\varepsilon h(x)$, then the following inequality holds*

$$\mathrm{PD}_{\lambda h}(x, v; z) \leq \tfrac{1}{2}\,\mathrm{M}_\lambda(x, v; z) + \lambda\varepsilon.$$

*Furthermore, it holds with equality when $\varepsilon = h(x) + h^*(v) - \langle x; v\rangle$.*

*Reciprocally, let $\sigma \geq 0$, $x, v, z \in \mathbb{R}^d$, if $\mathrm{PD}_{\lambda h}(x, v; z) \leq \sigma$ then,*

$$v \in \partial_{\varepsilon_v} h(x), \quad \textit{with } \varepsilon_v = \tfrac{\sigma}{\lambda} - \tfrac{1}{2\lambda}\,\mathrm{M}_\lambda(x, y; z).$$

Therefore, imposing an upper bound on the right hand side, automatically imposes a bound on the primal-dual gap. We list a series of criterion that were used in different works for quantifying the quality of some primal-dual pair $(x, v)$ for approximating the pair $(\mathrm{prox}_{\lambda h}(z), \mathrm{prox}_{h^*/\lambda}(z/\lambda))$. In all the criteria that follow, $\sigma$ denotes an error magnitude that we do not specify for now as we focus on the left hand side of the inexactness criteria.

- (Primal-dual inaccuracy, take I) The quality of a primal-dual pair $(x, v)$ for approximating the couple $(\mathrm{prox}_{\lambda h}(z), \mathrm{prox}_{h^*/\lambda}(z/\lambda))$ can be monitored by requiring

  $$\mathrm{PD}_{\lambda h}(x, v; z) \leq \sigma,$$

  to hold for some predefined $\sigma \geq 0$. Using Lemma 5.2.1, one can reformulate this requirement as $\exists \varepsilon \geq 0$: $v \in \partial_\varepsilon h(x)$ and $\frac{1}{2}\|x - z + \lambda v\|^2 + \lambda\varepsilon \leq \sigma$. This criterion is used among others in the hybrid approximate extragradient (HPE) framework [Solodov and Svaiter, 1999a, 2000a,b, 2001], in its inertial/accelerated versions [Monteiro and Svaiter, 2013, Boţ and Csetnek, 2015, Alves and Marcavillaca, 2019], or for forward-backward splittings [Millán and Machado, 2019, Bello-Cruz et al., 2020]. This criterion is generalized in the (monotone) operator world, through the notion of $\varepsilon$-enlargements [Burachik et al., 1998, 1997], generalizing the notion of $\varepsilon$-subdifferentials.

Stronger notions of primal-dual pairs can be obtained by coupling the primal and dual estimates, as follows.

- (Primal-dual inaccuracy, take II) The quality of a primal point $x$ for approximating $\mathrm{prox}_{\lambda h}(z)$ can be monitored by constructing an approximate dual point through (Moreau): $v = \frac{z-x}{\lambda}$ and requiring the corresponding primal-dual gap to satisfy

  $$\mathrm{PD}_{\lambda h}(x, \tfrac{z-x}{\lambda}; z) \leq \sigma.$$

  Note that this formulation can be rewritten as $\mathrm{PD}_{\lambda h}(x, \frac{z-x}{\lambda}; z) = \lambda h(x) + \lambda h^*(\frac{z-x}{\lambda}) - \lambda\langle x; \frac{x-z}{\lambda}\rangle \leq \sigma \Leftrightarrow \frac{z-x}{\lambda} \in \partial_{\sigma/\lambda} h(x)$, or equivalently $x = z - \lambda v$ with $v \in \partial_{\sigma/\lambda} h(x)$, or even in a dual form $v = \frac{z-u}{\lambda}$ with $u \in \partial_{\sigma/\lambda} h^*(v)$. This notion of inaccuracy was also used in quite a few works, see e.g., Lemaire [1992], Cominetti [1997] and more recently in Villa et al. [2013] and Salzo and Villa [2012, "approximation of type 2"].

- (Primal-dual inaccuracy, take III) The quality of a primal point $x$ for approximating $\mathrm{prox}_{\lambda h}(z)$ can be monitored by constructing an approximate dual point as $v = h'(x) \in \partial h(x)$ and by requiring

  $$\mathrm{PD}_{\lambda h}(x, h'(x); z) \leq \sigma.$$

  In this case, the criterion can be written as $\mathrm{PD}_{\lambda h}(x, h'(x); z) = \frac{1}{2}\|x - z + \lambda h'(x)\|^2 \leq \sigma$, which is equivalent to $x = z - \lambda v + \lambda e$ with $v \in \partial h(x)$ and $\frac{\lambda^2}{2}\|e\|^2 \leq \sigma$. This error criterion

was among the first to be used, see Rockafellar [1976b], and was later used in many works, see e.g., Burke and Qian [1999], Solodov and Svaiter [1999b, 2000a,b], Eckstein [1998], Alves et al. [2019], and Salzo and Villa [2012, "approximation of type 3"].

Among known methods for dealing with inexact proximal iterations, extra-gradient methods occupy an important place (see, e.g., the conceptual algorithm in Nemirovski [2004]). Intuitively, the idea is to compute some intermediate point $u \approx \mathrm{prox}_{\lambda h}(z)$, to evaluate some $u' \in \partial h(u)$ (or an $\epsilon$-subgradient version of it), and to use $x = z - \lambda u'$ as our working approximation of $\mathrm{prox}_{\lambda h}(z)$. Natural notions of inaccuracy applied on $u$ can also then directly be interpreted in terms of $x$, as follows.

- (Primal-dual inaccuracy, take IV) One way to interpret the hybrid proximal extra-gradient method [Solodov and Svaiter, 1999a] is that it measures the quality of a primal point $x$ for approximating $\mathrm{prox}_{\lambda h}(z)$ by requiring the existence of some other primal point $u$ satisfying

$$\mathrm{PD}_{\lambda h}(u, \tfrac{z-x}{\lambda}; z) \leq \sigma.$$

  Equivalently, one can write this condition as $\exists \varepsilon \geq 0$ and $\exists u \in \partial_\varepsilon h^*(\frac{z-x}{\lambda})$ such that $\frac{1}{2}\|u - x\|^2 + \lambda\varepsilon \leq \sigma$, which we can also explicitly rewrite in an extra-gradient format as: $x = z - \lambda u'$ with $u' \in \partial_\varepsilon h(u)$, $\frac{1}{2}\|u - z + \lambda u'\|^2 + \lambda\varepsilon \leq \sigma$ for some feasible $u$. In other words, it corresponds to obtain a $u \approx \mathrm{prox}_{\lambda h}(z)$ according to the primal-dual inaccuracy criterion (take I) on $u$, and to use $x = z - \lambda u'$ as the working approximation of $\mathrm{prox}_{\lambda h}(z)$.

- (Primal-dual inaccuracy, take V) A stronger version of the previous construction for measuring inaccuracy of $x$ consists in picking $u \in \partial h^*(\frac{z-x}{\lambda})$ and requiring

$$\mathrm{PD}_{\lambda h}(u, \tfrac{z-x}{\lambda}; z) = \tfrac{1}{2}\|x - u\|^2 \leq \sigma.$$

  In this setting, one can rewrite $x = z - \lambda u'$ with $u' \in \partial h(u)$ with $\frac{1}{2}\|x - u\|^2 \leq \sigma$. This condition was presented, and used, in Solodov and Svaiter [2000c] (though not exactly using this viewpoint). This corresponds to apply the primal-dual inaccuracy criterion (take III) on $u \approx \mathrm{prox}_{\lambda h}(z)$, and to use $x = z - \lambda u'$ as the working approximation of $\mathrm{prox}_{\lambda h}(z)$. This criterion is also used in Eckstein and Yao [2017] for relatively inexact Douglas-Rachford and ADMM.

Perhaps curiously, applying the same extra-gradient idea to primal-dual inaccuracy (take II), one recovers (take II) without any change.

One can then do the same exercise by requiring first-order optimality conditions to be approximatively satisfied. As previously explained, the corresponding notions of inexactness actually collapse with those based on primal-dual requirements as soon as either the dual variable is a subgradient of $h$ at the primal point $v \in \partial h(x)$, or equivalently when $x \in \partial h^*(v)$.

- (Primal-dual subgradient residual) Among the many possibilities for quantifying the quality of a primal-dual pair $(x, v)$ as an approximation of the solution $(\mathrm{prox}_{\lambda h}(z), \mathrm{prox}_{h^*/\lambda}(z/\lambda))$, one probably natural criterion is to require

$$\max\{\|x - z + \lambda v\|, \|v - \partial h(x)\|, \|x - \partial(h^*)(v)\|\} \leq \sigma.$$

  Another possibility is to require a positively weighted sum of those different terms to be small enough.

Note that one can design alternate criteria by performing conic combinations, intersections and unions of previous inaccuracy criteria. Finally, note that the choice of the most appropriate criterion depend on the application at hand (e.g., depending on the cost of obtaining an approximation satisfying the accuracy requirement, and on the cost of checking it).

**Remark 5.2.2.** *In practice, as soon as one can use a first-order (or higher-order) method for solving (P), (D) or (SP) there are often different ways to obtain primal-dual pairs (x,v) satisfying some primal-dual inexactness requirement. Depending on the application, $h^*$ and $\partial h^*$ might or might not be available, rendering some criterion irrelevant for that particular application. In particular, it is common that (P) can be solved approximately and one has access to elements of $\partial h(x)$. Then, criteria of the form $\mathrm{PD}_{\lambda h}(x, \partial h(x); z) \leq \sigma$ can be used, as in Alves et al. [2019].*

### 5.2.2 Abstract, generally non-observable, notions of inexactness

Some notions are more complicated to directly monitor in practice. However, they might allow modeling certain situations that are not covered by previous notions (such as dealing with possibly infeasible primal and dual solutions).

- (Purely primal (or dual) inaccuracy) The quality of an approximate $x \approx \mathrm{prox}_{\lambda h}(z)$ can be monitored by requiring $x$ to satisfy, for some $\sigma \geq 0$

$$\mathrm{PD}_{\lambda h}(x, \mathrm{prox}_{\frac{h^*}{\lambda}}(\tfrac{z}{\lambda}); z) = \Phi_p(x; z) - \Phi_p(\mathrm{prox}_{\lambda h}(z); z) \leq \sigma$$

  This notion is directly considered, e.g., in Auslender [1987], Schmidt et al. [2011], Lin et al. [2015, 2018], in Salzo and Villa [2012, "approximation of type 1"], and indirectly in other works (e.g., Güler [1992, Lemma 3.1]). Although it is mostly impractical (as it requires knowing the optimal value of the proximal subproblem), it can be verified indirectly via other error criterion (such as a primal-dual gap). In the same spirit, one could use purely dual requirements $\Phi_d(\mathrm{prox}_{h^*/\lambda}(z/\lambda); z) - \Phi_d(v; z)$.

- (Distance to the solution) A primal candidate $x \approx \mathrm{prox}_{\lambda h}(z)$ may be required to be close to $\mathrm{prox}_{\lambda h}(z)$. That is, for some $\sigma > 0$, one may require

$$\|x - \mathrm{prox}_{\lambda h}(z)\| \leq \sigma.$$

  This corresponds to control an approximate Moreau gap $\mathrm{M}_\lambda(x, v; z)$ with $v = \mathrm{prox}_{h^*/\lambda}(z/\lambda)$. This notion is also not new [Rockafellar, 1976b, Güler, 1992], and can also be verified indirectly, e.g. via $\frac{1}{2}\|x - \mathrm{prox}_{\lambda h}(z)\|^2 \leq \mathrm{PD}_{\lambda h}(x, \mathrm{prox}_{h^*/\lambda}(z/\lambda); z)$. Its dual version corresponding to $\|\mathrm{prox}_{h^*/\lambda}(z/\lambda) - v\|$, or a primal-dual notion as $\|x - \mathrm{prox}_{\lambda h}(z)\|^2 + \lambda^2\|\mathrm{prox}_{h^*/\lambda}(z/\lambda) - v\|^2$ could also be considered.

### 5.2.3 Absolute versus relative inaccuracies

Depending on algorithmic requirements, error tolerances might be specified in terms of absolute constants, or as functions of the state of the algorithm at hand. For example, a common situation is to choose some absolute constant $\sigma > 0$, and to require $\mathrm{PD}_{\lambda h}(x, v; z) \leq \sigma$, where $\sigma$ should typically be chosen as a decreasing function of the iteration counter. A standard alternative is to pick a relative type of accuracy requirement, such as $\mathrm{PD}_{\lambda h}(x, v; z) \leq \|x - z\|^2$. Both types of requirements are pretty standard, and were already stated in early developments on inexact proximal methods (see e.g., Rockafellar [1976b, condition (A) or (B)]. Relative versions often offer the advantage of being simpler to tune, sometimes at the cost of worse performances, see e.g., Lin et al. [2018].

## 5.3 Principled, and computer-assisted worst-case analyses

In this section, we show that a generic inexact proximal method can be analyzed using performance estimation problems. Those problems were introduced in Drori and Teboulle [2014] for analyzing fixed-step first-order methods for smooth convex optimization, and were extended in a few directions since then, see §"Computer-assisted analyses" in Section 5.1.2.

In short, we provide a principled approach to obtain rigorous worst-case guarantees and the corresponding proofs for a class of inexact proximal methods. The idea is to formulate the problem of performing a worst-case analysis as an optimization problem, which can be solved numerically. Feasible points to this problem correspond to matching examples (i.e., worst-case instances: functions and iterates) and feasible points to the dual problem correspond to worst-case guarantees (i.e., proofs). The possibility of solving those problems numerically essentially allows *sampling* worst-case examples and proofs for given problems and algorithmic parameters (for instance, step sizes and accuracy levels).

### 5.3.1 A class of inexact proximal methods

In this section we consider the minimization problem

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

with $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ (a closed, proper, and convex function) and define a class of approximate proximal methods for solving this problem, along with a principled way of analyzing them.

#### Fixed-step inexact proximal methods

Let $x_0 \in \mathbb{R}^d$ be an initial point, and let $\{\lambda_i\}_i$ be a sequence of nonnegative step sizes. When exact proximal computations are available, a natural class of methods can be described by

$$w_{k+1} = \operatorname{prox}_{\lambda_{k+1} h}\left(w_k - \sum_{i=1}^{k} \alpha_{k+1,i} v_i\right)$$

where $v_i \in \partial h(w_i)$ for $i = 1, \ldots, k$ and $\{\alpha_{i,j}\}_{ij}$ is a sequence of parameters. In this setting, the next iterate of the method is obtained as the result of the proximal operator of $h$ applied to the previous iterate plus a linear combination of previously encountered subgradients. It can be reformulated as

$$w_{k+1} = w_k - \sum_{i=1}^{k} \alpha_{k+1,i} v_i - \lambda_{k+1} v_{k+1}$$

where $v_{k+1} \in \partial h(w_{k+1})$, which corresponds to optimality conditions of the proximal subproblems.

We extend this class of algorithms for inexact proximal computations by introducing some error terms $\{e_i\}_i$ in the previous formulation.

$$w_{k+1} \approx \operatorname{prox}_{\lambda_{k+1} h}\left(w_k - \sum_{i=1}^{k} \alpha_{k+1,i} v_i - \sum_{i=0}^{k} \beta_{k+1,i} e_i\right)$$

where $v_i \in \partial h(w_i)$ for $i = 1, \ldots, k$ and $\{\alpha_{i,j}\}_{ij}$, $\{\beta_{i,j}\}_{ij}$ are sequences of parameters. In particular, $\{\beta_{i,j}\}_{ij}$ allows the method to take into account the errors made in previous proximal computations.

We disambiguate the $\approx$ notation by introducing an additional error term $e_{k+1}$ and define the class of *fixed-step inexact proximal methods* as

$$w_{k+1} = w_k - \sum_{i=1}^{k} \alpha_{k+1,i} v_i - \sum_{i=0}^{k} \beta_{k+1,i} e_i - \lambda_{k+1}(v_{k+1} + e_{k+1}), \tag{5.1}$$

where $v_{k+1} \in \partial h(w_{k+1})$. The error source in the proximal operation comes from the fact that $v_k + e_k$ does not necessarily belong to $\partial h(w_k)$. For modelling the error incurred in the proximal operations, in particular the discrepancy between $v_k + e_k$ an $\partial h(w_k)$, we are allowed to use all notions from previous sections. We abstract this modelling step by imposing on the iterates some (possibly vector) inequalities of the form

$$\mathrm{EQ}_k(w_0, \dots, w_k, v_0, \dots, v_k, e_0, \dots, e_k, h(w_0), \dots, h(w_k)) \leq 0. \tag{5.2}$$

For readability purposes, we abusively use $\mathrm{EQ}_k$ without explicitly instantiating the inputs in what follows.

In addition, all the inexactness criteria of Section 5.2 share a common structure which we refer to as "Gram-representable", as follows.

**Definition 5.3.1.** *A criterion (5.2) is **Gram-representable** if it is affine in $h(w_0), \dots, h(w_k)$ and in $\langle x; y \rangle$ for all $x, y \in \{w_i\}_{i\in[0,k]} \cup \{v_i\}_{i\in[0,k]} \cup \{e_i\}_{i\in[0,k]}$.*

Compared to Definition 3.2.1, this definition allows using affine combinations (i.e. possibility of adding constant terms). All methods in the form (5.1) subject to Gram-representable (5.2) can be analyzed in a principled way using the performance estimation procedure presented in the next section. Furthermore, all inaccuracy criterion presented in Section 5.2 are actually Gram-representable.

### Examples

Before going into the analyses, we provide a few examples of methods that fit into model (5.1) with Gram-representable models of the form (5.2). In all cases, we let $\{\lambda_k\}_k$ be a sequence of predefined step sizes.

- The vanilla *proximal minimization algorithm* is given by

  $$x_{k+1} = x_k - \lambda_{k+1} v_{k+1},$$

  with $v_{k+1} \in \partial h(x_{k+1})$. It fits in (5.1) with $\alpha_{i,j} = \beta_{i,j} = 0$, as well as $e_k = 0$ which can be transcribed into a Gram-representable (5.2).

- The *inexact proximal minimization algorithm* proposed in Rockafellar [1976a, Section 3] can be described by

  $$x_{k+1} = x_k - \lambda_{k+1}(v_{k+1} + e_{k+1}),$$

  with $v_{k+1} \in \partial h(x_{k+1})$, with the error term $e_{k+1}$ being controlled via either

  $$\|e_{k+1}\|^2 \leq \tfrac{\epsilon_{k+1}^2}{\lambda_{k+1}^2}, \quad \text{or} \quad \|e_{k+1}\|^2 \leq \tfrac{\delta_{k+1}^2}{\lambda_{k+1}^2}\|x_{k+1} - x_k\|^2$$

  for some appropriate sequence $\{\epsilon_k\}_k$ [Rockafellar, 1976a, Criterion (A')], or $\{\delta_k\}_k$ [Rockafellar, 1976a, Criterion (B')]. In both cases, the method fits into model (5.1) with $\alpha_{i,j} = \beta_{i,j} = 0$ and a Gram-representable (5.2). Depending on how we decide to control the error, we can either pick $\mathrm{EQ}_{k+1} = \|e_{k+1}\|^2 - \tfrac{\epsilon_{k+1}^2}{\lambda_{k+1}^2}$ or $\mathrm{EQ}_{k+1} = \|e_{k+1}\|^2 - \tfrac{\delta_{k+1}^2}{\lambda_{k+1}^2}\|x_{k+1} - x_k\|^2$.

Many known proximal methods rely on using the past first-order information for improving convergence guarantees of the sequence iterates.

- *Güler proximal point algorithm* [Güler, 1992, Section 6] is defined as follow given $\beta_0 > 0$, $y_0 = x_0 \in \mathbb{R}^d$ and $\{\lambda_k\}_k$ a sequence of positive step sizes

$$\left\{ \begin{array}{rcl} t_{k+1} & = & \frac{1+\sqrt{1+4t_k^2}}{2} \\ x_{k+1} & = & y_k - \lambda_{k+1} v_{k+1} \text{ with } v_{k+1} \in \partial h(x_{k+1}) \\ y_{k+1} & = & x_{k+1} + \frac{t_k - 1}{t_{k+1}}(x_{k+1} - x_k) + \frac{t_k}{t_{k+1}}(x_{k+1} - y_k) \end{array} \right.$$

  One can substitute the $y_{k+1}$ by $x_{k+2}+\lambda_{k+2}v_{k+2}$ and $y_k$ by $x_{k+1}+\lambda_{k+1}v_{k+1}$ in the last definition, which leads to

$$x_{k+2} = \left(1 + \tfrac{t_k - 1}{t_{k+1}}\right) x_{k+1} - \tfrac{t_k - 1}{t_{k+1}} x_k - \tfrac{t_k \lambda_{k+1}}{t_{k+1}} v_{k+1} - \lambda_{k+2} v_{k+2}.$$

  This allows to show recursively that the $\{x_k\}_k$ belong to the class (5.1). Indeed $x_1 = x_0 - \lambda_1 v_1$. Then suppose that $x_{k+1}$ has the form of (5.1), with $\beta_{k+1,i} = 0$ and $e_{k+1} = 0$, then

$$x_{k+2} = x_{k+1} - \tfrac{t_k - 1}{t_{k+1}} \left( \sum_{i=1}^{k} \alpha_{k+1,i} v_i + \lambda_{k+1} v_{k+1} \right) - \tfrac{t_k \lambda_{k+1}}{t_{k+1}} v_{k+1} - \lambda_{k+2} v_{k+2}.$$

  And we can identify $\alpha_{k+2,i} = \frac{t_k - 1}{t_{k+1}} \alpha_{k+1,i}$ for $i = 1 \dots k$ and $\alpha_{k+2,k+1} = \frac{t_k - 1}{t_{k+1}} \lambda_{k+1}$, as well as $\beta_{k+2,i} = 0$ and $e_{k+2} = 0$.

Other methods that fit in (5.1) with Gram-representable inexactness criterion (5.2) include the *hybrid approximate extragradient algorithm* [Solodov and Svaiter, 1999a] (details in Section 5.A), the *inexact accelerated proximal point algorithm IAPPA1* and *IAPPA2* [Salzo and Villa, 2012] (details in Section 5.A), *A-HPE* [Monteiro and Svaiter, 2013] (see details in Section 5.A), and *Catalyst* [Lin et al., 2015].

### 5.3.2 Computing worst-case guarantees

In this section, we provide a principled approach for performing worst-case analyses of fixed-step inexact proximal methods written in terms of (5.1) and (5.2). Let $N \in \mathbb{N}$ and $R \in \mathbb{R}^*$, for simplicity of the exposition, we only consider worst-case guarantees of type

$$h(w_N) - h(w_*) \leq C(N, R), \tag{5.3}$$

for all $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$, $w_* \in \operatorname{argmin}_x h(x)$, $w_0 \in \mathbb{R}^d$ such that $\|w_0 - w_*\|^2 \leq R^2$, and $d \in \mathbb{N}$. Our goal is then to compute values of $C(N, R)$, hopefully small and decreasing with $N$, for this inequality to be valid. This choice is made for simplicity purposes, and can be changed (e.g. Section 5.5); see discussions and examples in Taylor et al. [2017a,b].

Given a method in the form (5.1) (i.e., fixed $\{\alpha_{i,j}\}_{ij}$, $\{\beta_{i,j}\}_{ij}$) as well as inexactness criteria in the form (5.2) (i.e., fixed $\{\text{EQ}_i\}_i$), we formulate the problem of computing the smallest $C(N, R)$ such that (5.3) is valid. For doing that, we look for the worst problem instance for guarantees of type (5.3),

that is, a convex function on which $h(w_N) - h(w_*)$ is the largest possible when $\|w_0 - w_*\|^2 \leq R^2$

$$
\begin{aligned}
C(N,R) \geq \max_{\substack{d,h \\ w_*,w_0,\dots,w_N \in \mathbb{R}^d \\ v_0,\dots,v_N \in \mathbb{R}^d \\ e_0,\dots,e_N \in \mathbb{R}^d}} \quad & h(w_N) - h(w_*) \\
\text{s.t. } & h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d), \quad w_* \in \operatorname*{argmin}_x h(x) \\
& \|w_0 - w_*\|^2 \leq R^2 \\
& w_1, \dots, w_N \text{ satisfying (5.1)} \\
& \mathrm{EQ}_k \leq 0 \quad k = 0, \dots, N.
\end{aligned}
\tag{5.4}
$$

This type of problems is often referred to as a *performance estimation problem* (introduced in Drori and Teboulle [2014]). It is intrinsically infinite dimensional, as it contains a variable $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$. One possible way of dealing with this variable is to restrict ourselves to work with a discrete (or sampled) version of $h$. For doing that, we introduce a set $S$ containing sampled points of $h$, in the form $S = \{(w_i, v_i, h_i)\}_i$, and we reformulate the previous problem using the requirement $h_i = h(w_i)$, $v_i \in \partial h(w_i)$. In addition, (5.2) implies that the $\mathrm{EQ}_k$ are only described using $\{e_i\}_i$ and the elements of $S$ (we emphasize this by writing $\mathrm{EQ}_k(S, e)$), thus we can write

$$
\begin{aligned}
C(N,R) \geq \max_{\substack{d \\ S \subset \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R} \\ e_0,\dots,e_N \in \mathbb{R}^d}} \quad & h_N - h_* \\
\text{s.t. } & S = \{(w_i, v_i, h_i)\}_{i \in \{*,0,1,\dots,N\}} \\
& \exists h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d) : \; f = h(x), \; g \in \partial h(x) \quad \forall (x,g,f) \in S \\
& v_* = 0, \quad \|w_0 - w_*\|^2 \leq R^2 \\
& w_1, \dots, w_N \text{ satisfying (5.1)} \\
& \mathrm{EQ}_k(S, e) \leq 0 \quad k = 0, \dots, N.
\end{aligned}
\tag{5.5}
$$

Now, a key step is to rely on interpolation (also often referred to as extension) theorems for formulating the existence constraints in a tractable way. Such results can be formulated as follows (see e.g., Taylor et al. [2017c, Theorem 1]) :

$$
\begin{aligned}
\exists h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d) : \; f = h(x), \quad g \in \partial h(x) \quad \forall (x,g,f) \in S \\
\Leftrightarrow f' \geq f + \langle g; x' - x \rangle \quad \forall (x,g,f), (x',g',f') \in S.
\end{aligned}
\tag{5.6}
$$

It allows arriving to a nearly quadratic problem (still dependent on a dimension variable $d$).

$$
\begin{aligned}
\max_{\substack{d \\ S \subset \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R} \\ e_0,\dots,e_N \in \mathbb{R}^d}} \quad & h_N - h_* \\
\text{s.t. } & S = \{(w_i, v_i, h_i)\}_{i \in \{*,0,1,\dots,N\}} \\
& f' \geq f + \langle g; x' - x \rangle \quad \forall (x,g,f), (x',g',f') \in S \\
& v_* = 0, \quad \|w_0 - w_*\|^2 \leq R^2 \\
& w_1, \dots, w_N \text{ satisfying (5.1)} \\
& \mathrm{EQ}_k(S, e) \leq 0 \quad k = 0, \dots, N.
\end{aligned}
\tag{5.7}
$$

**Remark 5.3.2.** *Let us note that inexactness requirements for proximal operators are often formulated through $\varepsilon$-subdifferentials. In order to simplify the performance estimation problems, one can use appropriate interpolation conditions for directly incorporating $\varepsilon$-subdifferentials. Since this interpolation result is rather a trivial extension of regular convex interpolation (see e.g., Taylor et al. [2017c, Theorem 1]), we provide it in Section 5.B.*

The next section presents how problem (5.7) can be reformulated as linear semidefinite program when the $\text{EQ}_k$ are Gram-representable.

### 5.3.3 Semidefinite formulation

Let

$$H = [h_0 - h_*, \quad h_1 - h_*, \quad \dots \quad h_N - h_*] \in \mathbb{R}^{1\times(N+1)}$$

a flat vector containing function values and

$$\begin{aligned} &G = X^T X \succeq 0 \text{ with} \\ &X = [w_*, \ w_0, \ v_0, \ \dots \ v_N, \ e_0, \ \dots \ e_N] \in \mathbb{R}^{d\times(2N+4)} \end{aligned}$$

a Gram matrix of the vector variables of (5.7). For writing (5.7) as a semidefinite program, let us introduce base vectors $\mathbf{w}_k$, $\mathbf{v}_k$, and $\mathbf{e}_k$ in $\mathbb{R}^{2N+4}$ for conveniently selecting entries of $X$, and $\mathbf{h}_k$ in $\mathbb{R}^{N+1}$ for selecting entries of $H$, such that

$$\begin{aligned} w_k = X\mathbf{w}_k, v_k = X\mathbf{v}_k, \ e_k = X\mathbf{e}_k, \\ h_k = H\mathbf{h}_k. \end{aligned}$$

More precisely, we pick $\mathbf{w}_* = \mathbf{u}_1$, $\mathbf{v}_* = 0$, $\mathbf{w}_0 = \mathbf{u}_2$, $\mathbf{v}_k = \mathbf{u}_{k+3}$ $(k = 0, \dots, N)$, $\mathbf{e}_k = \mathbf{u}_{k+N+4}$ $(k = 0, \dots, N)$ with $\mathbf{u}_i$ the unit vector of $\mathbb{R}^{2N+4}$ with 1 at its $i$th component. For $\mathbf{w}_k$ $(k = 1, \dots, N)$, we use (5.1) and write

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \sum_{i=1}^{k} \alpha_{k+1,i}\mathbf{v}_i - \sum_{i=0}^{k} \beta_{k+1,i}\mathbf{e}_i - \lambda_{k+1}(\mathbf{v}_{k+1} + \mathbf{e}_{k+1}).$$

For function values, we define $\mathbf{h}_* = 0$ and $\mathbf{h}_k = \mathbf{u}_{k+1}$ $(k = 0, \dots, N)$ with $\mathbf{u}_i$ now in $\mathbb{R}^{N+1}$. In addition, when the constraints $\text{EQ}_k(S, e) \leq 0$ are Gram-representable, that is, each $\text{EQ}_k(S, e) \leq 0$ can be encoded as $m \in \mathbb{N}^*$ inequalities of the form $\text{EQ}_{k,i} = \text{Tr}(A_{k,i}G) + Ha_{k,i} \leq b_{k,i}$ with $A_{k,i} \in \mathbb{R}^{(2N+4)\times(2N+4)}$, $a_{k,i} \in \mathbb{R}^{N+1}$, $b_{k,i} \in \mathbb{R}$ and $i \in [1, m]$, (5.7) can finally be reformulated as

$$\begin{aligned} \max_{G\succeq 0,\, H} \ & H(\mathbf{h}_N - \mathbf{h}_*) \\ \text{s.t.} \ \ & 0 \ \geq H(\mathbf{h}_i - \mathbf{h}_j) + \mathbf{v}_i^T G(\mathbf{w}_j - \mathbf{w}_i) \quad \forall i, j \in \{*, 0, \dots, N\} \\ & R^2 \geq (\mathbf{w}_0 - \mathbf{w}_*)^T G(\mathbf{w}_0 - \mathbf{w}_*) \\ & 0 \ \geq -b_{i,j} + Ha_{i,j} + \text{Tr}(A_{i,j}G) \quad \forall i \in \{0, \dots, N\},\ j \in \{1, \dots, m\}, \end{aligned} \tag{5.8}$$

which is a linear semidefinite program. Feasible points correspond to discrete version of functions $h \in \mathcal{F}_{0,\infty}(Rd)$, which can be constructed through convex interpolation mechanisms [Taylor et al., 2017c].

**Remark 5.3.3.** *The $b_{i,j}$ terms in the inexactness criterion is here to take into account possible absolute (non-homogeneous) error terms (i.e., independent of the iterates).*

We have seen how to solve numerically the performance estimation problem (5.4) using a semidefinite reformulation (5.8). The objective of the next section is to present some duality arguments that allows to derive worst-case guarantees from feasible dual points of problem (5.8).

### 5.3.4 Recovering worst-case guarantees from dual solutions

The worst-case guarantees presented in the sequel were found using dual certificates (i.e., dual feasible points) of problem (5.8). In this section, we detail the relationship between such dual feasible points and traditional proofs not relying on semidefinite programming.

Let $\nu = \{\nu_{i,j}\}_{ij}$ be the nonnegative Lagrangian multipliers associated with the convex interpolation constraints and $\mu = \{\mu_{i,j}\}_{ij}$ the ones associated with inexactness constraints. We introduce the quantities

$$\begin{aligned}
\tilde{H}(\nu,\mu) &= \sum_{i,j\in\{*,0,\dots,N\}} \nu_{i,j}\left[(\mathbf{h}_i-\mathbf{h}_j)\right] + \sum_{\substack{i\in\{0,\dots,N\}\\ j\in\{1,\dots,m\}}} \mu_{i,j}a_{i,j},\\
\tilde{G}(\nu,\mu) &= \sum_{i,j\in\{*,0,\dots,N\}} \nu_{i,j}\left[(\mathbf{w}_j-\mathbf{w}_i)\mathbf{v}_i^T\right] + \sum_{\substack{i\in\{0,\dots,N\}\\ j\in\{1,\dots,m\}}} \mu_{i,j}A_{i,j},\\
\tilde{B}(\mu) &= \sum_{\substack{i\in\{0,\dots,N\}\\ j\in\{1,\dots,m\}}} \mu_{i,j}b_{i,j},
\end{aligned}$$

and the Lagrangian of problem (5.8) can be expressed as

$$\begin{aligned}
\mathcal{L}(G,H,\nu,\mu,\tau) =& H\left[\mathbf{h}_N-\mathbf{h}_*-\tilde{H}(\nu,\mu)\right] + \tau R^2 + \tilde{B}(\mu)\\
&- \operatorname{Tr}\left(\left[\tilde{G}(\nu,\mu)+\tau(\mathbf{w}_0-\mathbf{w}_*)(\mathbf{w}_0-\mathbf{w}_*)^T\right]G\right),
\end{aligned}$$

where $\tau \geq 0$ is the multiplier associated with the constraint on distance to optimality of the starting point.

Since the Lagrangian is linear in $G$ and $H$, maximizing with respect to $H$ and $G \succeq 0$ leads to the following dual function

$$\max_{\substack{G\succeq 0\\ H}} \mathcal{L}(G,H,\nu,\mu,\tau,D) = \begin{cases} \tau R^2+\tilde{B}(\mu) & \text{if } \mathbf{h}_N-\mathbf{h}_* = \tilde{H}(\nu,\mu) \text{ and}\\ & \frac{\tilde{G}(\nu,\mu)+\tilde{G}(\nu,\mu)^T}{2}+\tau(\mathbf{w}_0-\mathbf{w}_*)(\mathbf{w}_0-\mathbf{w}_*)^T \succeq 0\\ +\infty & \text{otherwise}\end{cases}$$

and the corresponding dual problem

$$\begin{aligned}
\min_{\substack{\tau\geq 0,\\ \nu\geq 0,\,\mu\geq 0}}\quad & \tau R^2+\tilde{B}(\mu)\\
\text{s.t. } & \mathbf{h}_N-\mathbf{h}_* = \tilde{H}(\nu,\mu)\\
& \tfrac{\tilde{G}(\nu,\mu)+\tilde{G}(\nu,\mu)^T}{2}+\tau(\mathbf{w}_0-\mathbf{w}_*)(\mathbf{w}_0-\mathbf{w}_*)^T \succeq 0.
\end{aligned} \tag{5.9}$$

Therefore, for any feasible dual point $(\nu,\mu,\tau)$, the following inequality is valid for all $G \succeq 0$, $H$

$$\mathcal{L}(G,H,\nu,\mu,\tau) \leq \tau R^2+\tilde{B}(\mu),$$

which can be rewritten as

$$\begin{aligned}\mathcal{L}(G, H, \nu, \mu, \tau) - \tau R^2 + \tilde{B}(\mu) &= H\left[\mathbf{h}_N - \mathbf{h}_* - \tilde{H}(\nu, \mu)\right] \\ &\quad - \operatorname{Tr}\left(\left[\tilde{G}(\nu, \mu) + \tau(\mathbf{w}_0 - \mathbf{w}_*)(\mathbf{w}_0 - \mathbf{w}_*)^T\right] G\right) \\ &\leq 0.\end{aligned}$$

Going back to the notations of problem (5.7) the previous inequality is equivalent to

$$\begin{aligned}h(w_N) - h_* - \tau\|w_0 - w_*\|^2 \leq &\sum_{i,j \in \{*,0,\dots,N\}} \nu_{i,j}[h(w_i) - h(w_j) + \langle v_i; w_j - w_i\rangle] \\ &+ \sum_{\substack{i \in \{0,\dots,N\} \\ j \in \{1,\dots,m\}}} \mu_{i,j}\mathrm{EQ}_{i,j} + \tilde{B}(\mu) \\ &\leq \tilde{B}(\mu),\end{aligned} \tag{5.10}$$

the last inequality comes from the fact that the dual variables are (element-wise) nonnegative, $v_i \in \partial h(w_i)$, and $\mathrm{EQ}_{i,j} \leq 0$. Therefore, we get that

$$h(w_N) - h_* \leq \tau\|w_0 - w_*\|^2 + \tilde{B}(\mu).$$

Thus, obtaining admissible dual points $\tau$, $\nu$, $\mu$ of problem (5.8) provides a way of combining interpolation inequalities and inexactness criterion such that (5.10) is valid (examples of proofs relying on this mechanism can be found e.g., Klerk et al. [2017], Lieder [2020], Taylor and Bach [2019]).

**Remark 5.3.4.** *The quantity $\tau R^2 + \tilde{B}(\mu)$ is always an upper-bound on $C(N, R)$ when $(\tau, \nu, \mu)$ is a feasible point of (5.9). Furthermore, under mild conditions for zero duality gap to occur (e.g., when Slater's condition holds for the primal problem), the smallest possible $C(N, R)$ satisfying (5.3) is exactly equal to $\tau_* R^2 + \tilde{B}(\mu_*)$ where $(\tau_*, \nu_*, \mu_*)$ is an optimal solution to (5.9).*

**Remark 5.3.5.** *When there is no absolute error in the proximal computations (i.e., $b_{i,j} = 0$) which corresponds to inequalities $EQ_k$ that are 1-homogeneous in function values and 2-homogeneous in vector variables, then $\tilde{B}(\mu) = 0$ and the convergence guarantees have the standard form $h(w_N) - h_* \leq \tau\|w_0 - w_*\|^2$. In addition, we notice that solutions to the dual problem (5.9) are independent of $R$ and the optimal objective value is proportional to $R^2$.*

In the rest of the chapter we use this framework to analyze some optimization methods with inexact proximal computations under different inexactness criteria.

### 5.3.5 Numerical examples

In this section we instantiate various inexact proximal minimization methods and exhibits numerical worst-case guarantees using the framework of Section 5.3.

#### A simple relatively inexact proximal point method

The *inexact proximal minimization algorithm* with fixed step size presented in Section 5.3.1 corresponds to updates $w_{k+1} = w_k - \lambda(v_{k+1} + e_{k+1})$, with $v_{k+1} \in \partial h(w_{k+1})$, where we impose a criterion

of the form (Primal-dual inaccuracy, take III) that is controlled relatively by the distance between two consecutive iterates. This corresponds to

$$\mathrm{EQ}_{k+1} = \|e_{k+1}\|^2 - \tfrac{\sigma^2}{\lambda}\|w_{k+1} - w_k\|^2 \leq 0$$

for a fixed $\sigma \geq 0$. In this setting, problem (5.8) is of the form

$$\begin{aligned} \max_{G \succeq 0,\, H} \; & H(\mathbf{h}_N - \mathbf{h}_*) \\ \text{s.t.} \; & 0 \geq H(\mathbf{h}_i - \mathbf{h}_j) + \mathbf{v}_i^T G(\mathbf{w}_j - \mathbf{w}_i) \quad \forall i,j \in \{*, 0, \dots, N\} \\ & R^2 \geq (\mathbf{w}_0 - \mathbf{w}_*)^T G(\mathbf{w}_0 - \mathbf{w}_*) \\ & 0 \geq \mathbf{e}_i^T G \mathbf{e}_i - \tfrac{\sigma^2}{\lambda^2}(\mathbf{w}_i - \mathbf{w}_{i-1})^T G(\mathbf{w}_i - \mathbf{w}_{i-1}) \quad \forall i \in \{1, \dots, N\}, \end{aligned} \tag{5.11}$$

using notations of Section 5.3.3.

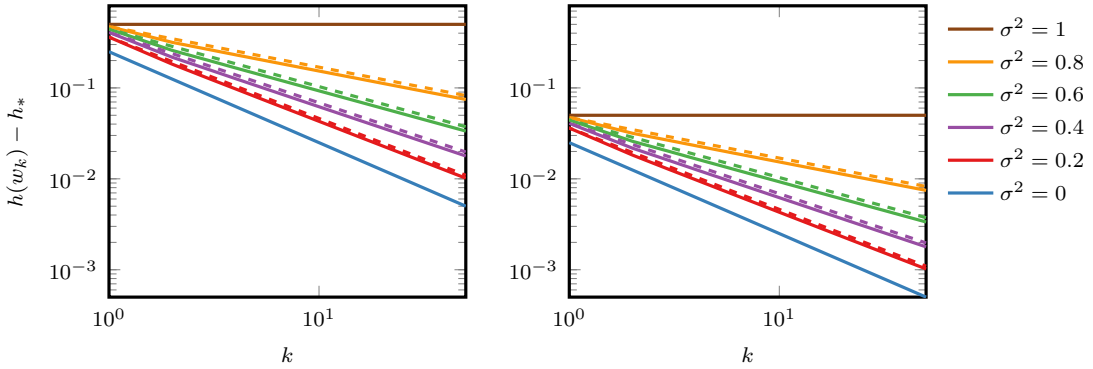


Figure 5.3.1: Numerical worst-case guarantees on $h(w_k) - h_*$ with initial condition $\|w_0 - w_*\|^2 \leq 1$, as function of $k$ (obtained by solving semidefinite programs (5.11)) for the relatively inexact proximal point algorithm of Section 5.3.5, with parameters $\lambda = 1$ (left), and $\lambda = 10$ (right). The dashed lines are empirical upper bounds of the form $(1+\sigma)/(4\lambda k^{\sqrt{1-\sigma^2}})$ which we plotted for reference. The semidefinite programs were solved through Löfberg [2004] and Mosek [2010].

One can now solve (5.11) numerically, for different values of $\sigma$, $\lambda$ and $R$, using standard semidefinite solvers (see e.g.; Mosek [2010], Sturm [1999]). The corresponding numerical worst-case bounds are provided in Figure 5.3.1 for different parameter values. Based on numerical experiments, we conjecture the expression $R^2(1+\sigma)/(4\lambda N^{\sqrt{1-\sigma^2}})$ to be a valid $C(N,R)$. For this example, we do not have a proof for this bound, as the algebra involved in obtaining an analytical form of a dual feasible point (as described in Section 5.3.4) turned out to be quite complicated in our trials on this simple method.

This example illustrates how we can use the performance estimation approach to compute worst-case bounds numerically, even when rigorous analytical proofs seem out of reach.

#### Inexact accelerated proximal point algorithms *IAPPA*

As detailed in Section 5.A, *IAPPA1* and *IAPPA2* from Salzo and Villa [2012, Section 5] fit into the formalism of Section 5.3. In particular, one can apply Section 5.3.3 to compute numerical worst-case guarantees, as provided in Figure 5.3.2.

Regarding the numerical experiments, note that it might be delicate to deduce asymptotic convergence convergence rates by looking only at about a hundred of iterations. This is the limiting part of this

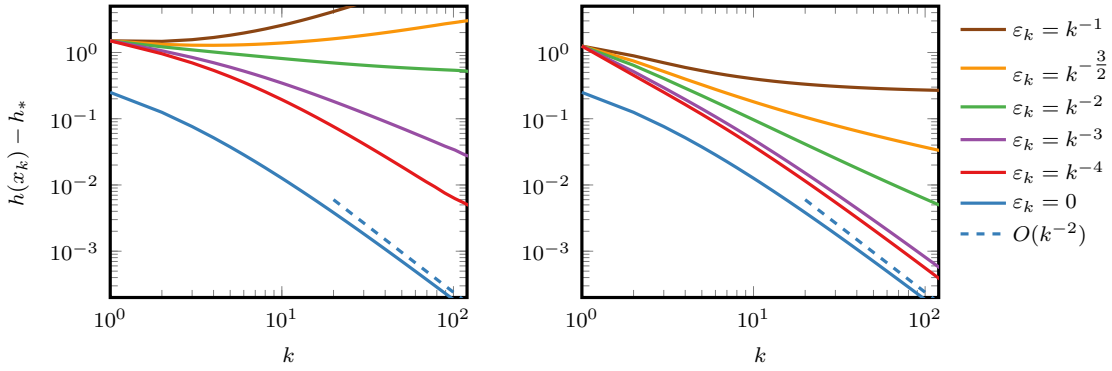


Figure 5.3.2: Numerical worst-case guarantees on $h(x_k) - h_*$ with initial condition $\|w_0 - w_*\|^2 \leq 1$, as function of $k$ for *IAPPA1* (left) and *IAPPA2* (right), with constant step size equal to 1 and $(\varepsilon_k)_k$ the sequence of parameters controlling the primal-dual gap values. We observe that for *IAPPA1* (left), the cases $\varepsilon_k = k^{-4}$ (red) and $\varepsilon_k = k^{-3}$ (purple) seems to decrease as $O(k^{-1})$ as stated in Salzo and Villa [2012, Theorem 4]. For *IAPPA2* (right), Salzo and Villa [2012, Theorem 6] states that $\varepsilon_k = k^{-4}$ (red) and $\varepsilon_k = k^{-3}$ (purple) curves of Figure 5.3.2 (right) should exhibit a convergence in $O(k^{-2})$, as observed. More iterations might be needed to observe the same phenomenon for the $\varepsilon_k = k^{-2}$ (green).

approach: the number of constraints in the semidefinite problems defined in Section 5.3.3 grows with the square of the number of iterations, which limits our capabilities of solving the corresponding problem. However, we can still make some observations, and sometimes deduce proofs (see Section 5.3.4).

Let us compare numerical worst-case guarantees in Figure 5.3.2 with convergence theorems [Salzo and Villa, 2012, Theorem 4, Theorem 6] for *IAPPA1* and *IAPPA2*. First note that Salzo and Villa [2012, Theorem 4] states that primal gap in *IAPPA1* converges to 0 as soon as $\varepsilon_k = O(k^{-q})$ with $q > \frac{3}{2}$, which is compatible with numerical experiments in Figure 5.3.2 (left). Reciprocally, it does seem that $q \leq \frac{3}{2}$ the worst-case guarantee does not converge to 0, apparently tightening Salzo and Villa [2012, Theorem 4]. Similar observations hold for algorithm *IAPPA2* (which involves a stricter inexactness requirements) with convergence of the primal gap for $q > 1/2$.

**Accelerated hybrid proximal Extragradient method (A-HPE)**

As detailed in Section 5.A, the *A-HPE* method from Monteiro and Svaiter [2013, Section 3] also fits into the formalism of Section 5.3. In particular, one can apply Section 5.3.3 to compute numerical worst-case guarantees that we provide in Figure 5.3.3.

The numerical bounds on $h(y_k) - h_*$ that we obtain in Figure 5.3.3 for $\sigma = 1$ seems to match exactly the analytical bound $\frac{\|w_0 - w_*\|^2}{2A_k}$ provided in Monteiro and Svaiter [2013, Theorem 3.6]. We further observe that numerical worst-case guarantees for all $\sigma \in [0, 1]$ tend to match with this analytical bound when the number of iterations gets larger.

In the next section we describe an optimized relatively inexact proximal point method with worst-case behavior derived from a dual feasible point, as previously described in Section 5.3.4.

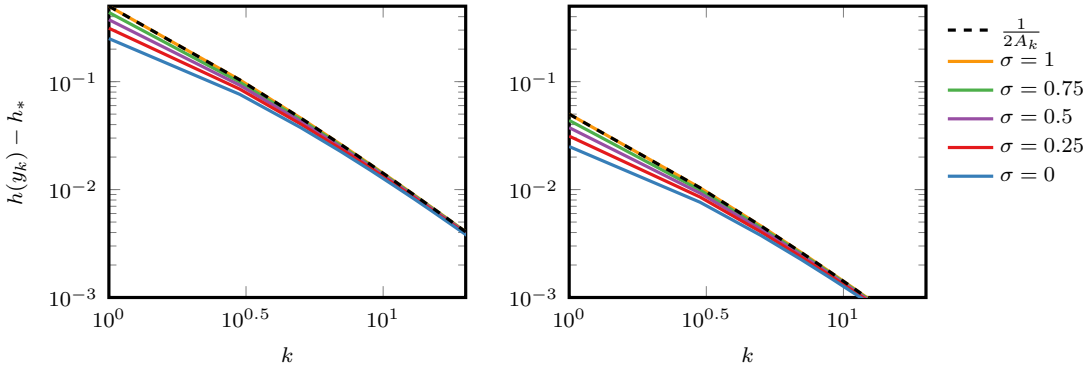


Figure 5.3.3: Numerical worst-case guarantees on $h(y_k) - h_*$ with initial condition $\|w_0 - w_*\|^2 \leq 1$, as function of $k$ for the *A-HPE* method with constant step size $\lambda = 1$ (left), and $\lambda = 10$ (right). The dashed curve corresponds to a theoretical upper bound on the primal gap from Monteiro and Svaiter [2013, Theorem 3.6].

## 5.4 An optimized relatively inexact proximal point algorithm

In this section we use the framework detailed in the Section 5.3 for designing an inexact proximal minimization algorithm with optimized worst-case performances. Similar to (5.1), provided sequences of step sizes $\{\lambda_i\}_i$ and parameters $\{\alpha_{i,j}\}_{ij}$, $\{\beta_{i,j}\}_{ij}$, we consider iterates of the form

$$x_{k+1} = x_k - \sum_{i=1}^{k} \alpha_{k+1,i} g_i - \sum_{i=1}^{k} \beta_{k+1,i} e_i - \lambda_{k+1}(g_{k+1} + e_{k+1}), \tag{5.12}$$

and impose an inexactness criterion of the form

$$\mathrm{PD}_{\lambda_k h}(x_k, g_k; x_k + \lambda_k(g_k + e_k)) \leq \tfrac{\sigma^2}{2}\|\lambda_k(g_k + e_k)\|^2, \quad k \geq 1.$$

This class of methods actually fits into (5.1) and (5.2), as shown in the next section.

Note that as mentioned in Remark 5.3.5, in the absence of non-homogeneous error terms in the inexactness criteria (which is the case here) and given methods parameters, provable worst-case guarantees derived from dual certificates are independent of the bound on the initial distance to optimality $R$. Therefore, we fix $R = 1$ in the performance estimation problems studied in this section for simplicity.

In order to find parameters $\{\alpha_{i,j}\}_{ij}$ and $\{\beta_{i,j}\}_{ij}$ that provide the smallest possible worst-case guarantees on $h(x_N) - h_*$ after $N \in \mathbb{N}^*$ iterations, we define

$$\begin{aligned} W\left(\{\alpha_{i,j}\}_{ij}, \{\beta_{i,j}\}_{ij}\right) := \max_{\substack{d,h \\ x_*, x_0, \ldots, x_N \in \mathbb{R}^d \\ g_0, \ldots, g_N \in \mathbb{R}^d \\ e_0, \ldots, e_N \in \mathbb{R}^d}} & \quad h(x_N) - h(x_*) \\ \text{s.t. } & h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d), \quad x_* \in \operatorname*{argmin}_x h(x) \\ & \|x_0 - x_*\|^2 \leq 1 \\ & x_1, \ldots, x_N \text{ satisfying (5.12)} \\ & \mathrm{PD}_{\lambda_k h}(x_k, g_k; x_k + \lambda_k(g_k + e_k)) \\ & \leq \tfrac{\sigma^2}{2}\|\lambda_k(g_k + e_k)\|^2 \quad k = 1, \ldots, N, \end{aligned} \tag{5.13}$$

and wish to solve the following problem

$$\underset{\{\alpha_{i,j}\}_{ij},\{\beta_{i,j}\}_{ij}}{\operatorname{argmin}} \quad W\left(\{\alpha_{i,j}\}_{ij},\{\beta_{i,j}\}_{ij}\right). \tag{5.14}$$

The rest of the section is organized as follow. First, we reformulate the method (5.12) and problem (5.13) for fitting into the setting and notations of Section 5.3. Then, since solving (5.14) exactly is out of reach in general, we detail a procedure to obtain feasible points (i.e., methods parameters) with optimized objective value. Finally, we present the method obtained from this choice of parameters together with its worst-case analysis.

### 5.4.1 Reformulation as fixed-step inexact proximal methods

The difference between (5.12) and (5.1) lies in the fact that we do not enforce $g_i \in \partial h(x_i)$ in the first model. In order to cast (5.12) into (5.1), we can define iterates as

$$\left\{\begin{array}{rcl} w_{2k+1} & = & w_{2k} - e_{2k} \\ w_{2k+2} & = & w_{2k} - \sum_{i=1}^{k} \alpha_{k+1,i} v_{2i-1} - \sum_{i=1}^{k} \beta_{k+1} e_{2i-1} - \lambda_{k+1}(v_{2k+1} + e_{2k+1}), \end{array}\right. \tag{5.15}$$

with $v_i \in \partial h(w_i)$, which fits into (5.1).

The inexactness requirements are then, $\mathrm{EQ}_0 = 0$, $\mathrm{EQ}_{2k+1} = 0$ and

$$\mathrm{EQ}_{2k+2} = \mathrm{PD}_{\lambda_{k+1}h}(w_{2k+2}, v_{2k+1}, w_{2k+2} + \lambda_{k+1}(v_{2k+1} + e_{k+1})) - \tfrac{\sigma^2}{2}\|\lambda_{k+1}(v_{2k+1} + e_{2k+1})\|^2.$$

Since $v_{2k+1} \in \partial h(w_{2k+1})$, we can write

$$h^*(v_{2k+1}) = \langle v_{2k+1}; w_{2k+1}\rangle - h(w_{2k+1}),$$

in the primal-dual gap and thus

$$\begin{aligned} \mathrm{EQ}_{2k+2} = {} & \tfrac{\lambda_{k+1}}{2}\|e_{2k+1}\|^2 - \tfrac{\lambda_{k+1}\sigma^2}{2}\|v_{2k+1} + e_{2k+1}\|^2 + h(w_{2k+2}) - h(w_{2k+1}) \\ & - \langle v_{2k+1}; w_{2k+2} - w_{2k+1}\rangle. \end{aligned}$$

which is Gram-representable. Finally, we can identify iterates $\{w_{2k}\}_k$ with the $\{x_k\}_k$ from (5.12) and we have

$$\begin{aligned} W\left(\{\alpha_{i,j}\}_{ij},\{\beta_{i,j}\}_{ij}\right) = \max_{\substack{d,h \\ w_*,w_0,\dots,w_{2N}\in\mathbb{R}^d \\ v_0,\dots,v_{2N}\in\mathbb{R}^d \\ e_0,\dots,e_{2N}\in\mathbb{R}^d}} \quad & h(w_{2N}) - h(w_*) \\ \text{s.t. } & h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d), \quad w_* \in \operatorname*{argmin}_x h(x) \\ & \|w_0 - w_*\|^2 \leq 1 \\ & w_1,\dots,w_{2N} \text{ satisfying (5.15)} \\ & \mathrm{EQ}_k \leq 0 \quad k = 0,\dots,N. \end{aligned} \tag{5.16}$$

In the following we first give a high level overview of how we can use a relaxation of (5.16) inside the minimization problem (5.14) to get a feasible point with optimized worst-case bound, and then present the algorithm obtained with this choice of optimized parameters together with its sharp convergence guarantees (sharp in the sense that given $\{\lambda_k\}_k$ and $N$ we can find a function for which the worst-case guarantee is attained exactly).

### 5.4.2 Obtaining optimized parameters

Problem (5.14) can be formulated as a linear minimization problem under a bilinear matrix inequality, which is NP-hard in general (see e.g., Toker and Ozbay [1995]). Thus, we approximate it by using a technique similar to that of Drori and Teboulle [2014], Kim and Fessler [2016], which consists in four steps.

(i) Find a suitable relaxation of the inner maximization problem (5.16) i.e., only keep a subset of the interpolation constraints. This relaxation is chosen by a numerical trial and error procedure.

(ii) Dualize the relaxed semidefinite formulation of the inner maximization problem to obtain a two-level minimization problem.

(iii) Use a change of variable similar to that in Drori and Teboulle [2014, Section 5] to remove non-linear terms in the bilinear semidefinite problem obtained at the previous step.

(iv) Retrieve a feasible point of (5.14) from the solution of the problem obtained in step (iii).

The final choice for the relaxation of (5.16) consisted in using only the following interpolation inequalities:

- convexity inequality between $w_{2k}$ and $w_{2k+1}$

$$h(w_{2k}) \geq h(w_{2k+1}) + \langle v_{2k+1}; w_{2k} - w_{2k+1} \rangle,$$

- convexity inequality between $w_*$ and $w_{2k+1}$

$$h(w_*) \geq h(w_{2k+1}) + \langle v_{2k+1}; w_* - w_{2k+1} \rangle,$$

- convexity inequality between $w_{2k+2}$ and $w_{2k+1}$

$$h(w_{2k+2}) \geq h(w_{2k+1}) + \langle v_{2k+1}; w_{2k+2} - w_{2k+1} \rangle,$$

along with inexactness conditions $\text{EQ}_k$. Those are exactly the inequalities used in the proof in next section.

More precisely, step (i) consisted in replacing $W\left(\{\alpha_{i,j}\}_{ij}, \{\beta_{i,j}\}_{ij}\right)$ in problem (5.14) by a relaxed version $U\left(\{\alpha_{i,j}\}_{ij}, \{\beta_{i,j}\}_{ij}\right)$ defined in its semidefinite form as follow

$$\begin{aligned}
&U\left(\{\alpha_{i,j}\}_{ij}\ , \{\beta_{i,j}\}_{ij}\right) := \\
&\quad \max_{G \succeq 0,\, H} \ H(\mathbf{h}_{2N} - \mathbf{h}_*) \\
&\qquad \text{s.t. } 0 \geq H(\mathbf{h}_{2i+1} - \mathbf{h}_{2i}) + \mathbf{v}_{2i+1}^T G(\mathbf{w}_{2i} - \mathbf{w}_{2i+1}) \quad \forall i \in \{1, \dots, N-1\} \\
&\qquad\quad 0 \geq H(\mathbf{h}_{2i-1} - \mathbf{h}_*) + \mathbf{v}_{2i-1}^T G(\mathbf{w}_* - \mathbf{w}_{2i-1}) \quad \forall i \in \{1, \dots, N\} \\
&\qquad\quad 0 \geq H(\mathbf{h}_{2i} - \mathbf{h}_{2i-1}) + \mathbf{v}_{2i-1}^T G(\mathbf{w}_{2i} - \mathbf{w}_{2i-1}) \quad \forall i \in \{1, \dots, N\} \\
&\qquad\quad 1 \geq (\mathbf{w}_0 - \mathbf{w}_*)^T G(\mathbf{w}_0 - \mathbf{w}_*) \\
&\qquad\quad 0 \geq \tfrac{\lambda_i}{2} \mathbf{e}_{2i-1}^T G \mathbf{e}_{2i-1} + H(\mathbf{h}_{2i} - \mathbf{h}_{2i-1}) - \mathbf{v}_{2i-1}^T G(\mathbf{w}_{2i} - \mathbf{w}_{2i-1}) \\
&\qquad\qquad - \tfrac{\sigma^2 \lambda_i}{2} (\mathbf{e}_{2i-1} + \mathbf{v}_{2i-1})^T G(\mathbf{e}_{2i-1} + \mathbf{v}_{2i-1}) \quad \forall i \in \{1, \dots, N\},
\end{aligned} \tag{5.17}$$

Then, step (ii) consisted in dualizing the maximization problem as seen in Section 5.3.4. From there, we search for parameters $\{\alpha_{i,j}\}_{ij}, \{\beta_{i,j}\}_{ij}$ that minimize the optimal value of the dual of (5.17). This is a minimization problem in $\{\alpha_{i,j}\}_{ij}, \{\beta_{i,j}\}_{ij}$ and in the dual variables of (5.17), that contains bilinear terms.

In step (iii), the bilinear terms in the minimization problem of step (ii) are replaced by new variables, producing a linear semidefinite program that can be solved efficiently.

Finally, in the last step, we retrieve parameters $\{\tilde{\alpha}_{i,j}\}_{ij}$ and $\{\tilde{\beta}_{i,j}\}_{ij}$ from the solutions of the linear semidefinite program of step (iii). Note that the relaxation step (i) is chosen so that step (iv) is achievable.

In the following, we describe the algorithm obtained from the choice of parameters $\{\tilde{\alpha}_{i,j}\}_{ij}$, $\{\tilde{\beta}_{i,j}\}_{ij}$.

### 5.4.3 Algorithm and convergence guarantees

**Optimized relatively inexact proximal point algorithm (ORI-PPA)**

**Input:**

- Objective function: $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$.
- Initial guess: $x_0 \in \mathbb{R}^d$.
- Inexactness parameter: $\sigma \in [0, 1]$.

**Initialization:**

$z_0 = x_0$, $A_0 = 0$,

**Run:**

For $k = 0, 1, \ldots$:

$$
\begin{aligned}
&\text{Choose } \lambda_{k+1} \geq 0 \\
&A_{k+1} = A_k + \tfrac{\lambda_{k+1} + \sqrt{4\lambda_{k+1}A_k + \lambda_{k+1}^2}}{2} \\
&y_k = x_k + \tfrac{\lambda_{k+1}}{A_{k+1} - A_k}(z_k - x_k) \\
&\Big[\text{Obtain } (x_{k+1}, g_{k+1}) \approx \Big(\operatorname{prox}_{\lambda_{k+1}h}(y_k), \operatorname{prox}_{h^*/\lambda_{k+1}}(\tfrac{y_k}{\lambda_{k+1}})\Big) \\
&\text{which satisfies } \mathrm{PD}_{\lambda_{k+1}h}(x_{k+1}, g_{k+1}; y_k) \leq \tfrac{\sigma^2}{2}\|x_{k+1} - y_k\|^2\Big] \\
&z_{k+1} = z_k - \tfrac{2(A_{k+1} - A_k)}{1+\sigma} g_{k+1}
\end{aligned}
\qquad \text{(ORI-PPA)}
$$

End For

**Output:** $x_{k+1}$

Perhaps luckily, it turns out that the parameters $\{\tilde{\alpha}_{i,j}\}_{ij}$ and $\{\tilde{\beta}_{i,j}\}_{ij}$ obtained from the four step procedure of Section 5.4.2 follow recursive equations allowing to rewrite iterations (5.1) under a

more compact form as presented in Algorithm (ORI-PPA) above. As mentioned earlier the iterates $\{x_k\}_k, \{g_{k+1}\}_k$ corresponds to the $\{w_{2k}\}_k, \{v_{2k+1}\}_k$ from (5.15) using$\{\tilde{\alpha}_{i,j}\}_{ij}$ and $\{\tilde{\beta}_{i,j}\}_{ij}$.

The Algorithm (ORI-PPA) is actually almost the same as the A-HPE algorithm from Monteiro and Svaiter [2013] (in particular definitions of sequences $\{y_k\}_k$, $\{z_k\}_k$ are the same when $\sigma = 1$). The main differences reside in the inexactness criterion, as we prefer to use primal-dual formulation rather than using $\varepsilon$-subgradients, and in the fact that (ORI-PPA) uses explicitly the inexactness level $\sigma$ in its step sizes. This last difference allows to improve the worst-case guarantee by a constant factor $\frac{1+\sigma}{2} \leq 1$ compared to Monteiro and Svaiter [2013, Theorem 3.6].

Perhaps surprisingly, this method reduces to that of Güler [Güler, 1992, Section 6] when using exact proximal operations ($\sigma = 0$) and constant step size, although the current method was obtained by crude numerical optimization of its parameters (see Section 5.C for details).

Solving numerically the dual of (5.17) allows to obtain rather simple analytical form for the optimal dual variables. We use these multipliers as in Section 5.3.4, to prove the following theorem.

**Theorem 5.4.1.** *Let $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$, a sequence of step sizes $\{\lambda_k\}_k$ with $\lambda_k > 0$, and $\sigma \in [0, 1]$. For any starting point $x_0 \in \mathbb{R}^d$, $N \geq 1$, the iterates of* (ORI-PPA) *satisfy*

$$h(x_N) - h(x_*) \leq \frac{(1+\sigma)\|x_0 - x_*\|^2}{4A_N},$$

*with $x_* \in \operatorname{argmin}_x h(x)$. Furthermore, this bound is tight: for all $\{\lambda_k\}_k$ with $\lambda_k > 0$, $\sigma \in [0, 1]$, $d \in \mathbb{N}$, $x_0 \in \mathbb{R}^d$, and $N \in \mathbb{N}$, there exists $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ such that this bound is achieved with equality.*

*Proof.* For the sake of clarity, we present the proof using notations of (ORI-PPA), although the proof was found via the SDP formulation (5.17).

We start with the case $\sigma \in (0, 1]$. The case $\sigma = 0$ is considered afterward as it requires a slightly different treatment.

In the following we denote by $u_{k+1}$ a point satisfying $u_{k+1} \in \partial h^*(g_{k+1})$ or equivalently $g_{k+1} \in \partial h(u_{k+1})$. These $u_{k+1}$ can be identified with the $w_{2k+1}$ in (5.15).

Consider the following inequalities with their corresponding weights :

- convexity between $x_k$ and $u_{k+1}$ with weight $\nu_{k,k+1} = \frac{A_k}{1+\sigma}$
  (for $k = 1, \ldots, N-1$)
  $$h(x_k) \geq h(u_{k+1}) + \langle g_{k+1}; x_k - u_{k+1} \rangle,$$
- convexity between $x_*$ and $u_k$ with weight $\nu_{*,k} = \frac{A_k - A_{k-1}}{1+\sigma}$
  (for $k = 1, \ldots, N$)
  $$h(x_*) \geq h(u_k) + \langle g_k; x_* - u_k \rangle,$$
- convexity between $x_k$ and $u_k$ with weight $\nu_{k,k} = \frac{A_k(1-\sigma)}{\sigma(1+\sigma)}$
  (for $k = 1, \ldots, N$)
  $$h(x_k) \geq h(u_k) + \langle g_k; x_k - u_k \rangle,$$
- approximation requirement on $x_k$ with weight $\nu_k = \frac{A_k}{\sigma(1+\sigma)}$
  (for $k = 1, \ldots, N$)
  $$\frac{\sigma^2}{2\lambda_k}\|x_k - y_{k-1}\|^2 \geq \frac{1}{2\lambda_k}\|x_k - y_{k-1} + \lambda_k g_k\|^2 + h(x_k) - h(u_k) - \langle g_k; x_k - u_k \rangle.$$

By linearly combining the previous inequalities, with their corresponding weights (which are nonnegative), we arrive to the following valid inequality:

$$\begin{aligned}
&\sum_{k=1}^{N-1} \nu_{k,k+1} h(x_k) + \sum_{k=1}^{N} \nu_{*,k} h(x_*) + \sum_{k=1}^{N} \nu_{k,k} h(x_k) + \sum_{k=1}^{N} \nu_k \tfrac{\sigma^2}{2\lambda_k} \|x_k - y_{k-1}\| \\
&\quad \geq \sum_{k=1}^{N-1} \nu_{k,k+1} [h(u_{k+1}) + \langle g_{k+1}; x_k - u_{k+1} \rangle] \\
&\qquad + \sum_{k=1}^{N} \nu_{*,k} [h(u_k) + \langle g_k; x_* - u_k \rangle] + \sum_{k=1}^{N} \nu_{k,k} [h(u_k) + \langle g_k; x_k - u_k \rangle] \\
&\qquad + \sum_{k=1}^{N} \nu_k [\tfrac{1}{2\lambda_k} \|x_k - y_{k-1} + \lambda_k g_k\|^2 + h(x_k) - h(u_k) - \langle g_k; x_k - u_k \rangle].
\end{aligned}$$

Substituting $x_k$ by its expression in (5.28), a reasonable amount of work (see Section 5.D) allows reformulating this inequality exactly as

$$\begin{aligned}
\tfrac{A_N}{1+\sigma}(h(x_N) - h_*) \leq & \tfrac{1}{4}\|x_0 - x_*\|^2 - \tfrac{1}{4}\|x_* - x_0 + \tfrac{2}{1+\sigma} \sum_{i=1}^{N} (A_i - A_{i-1}) g_i\|^2 \\
& - \tfrac{1-\sigma}{2\sigma} \sum_{i=1}^{N} A_i \lambda_i \|\tfrac{y_{i-1} - \lambda_i g_i - x_i}{\lambda_i} + \tfrac{\sigma}{1+\sigma} g_i\|^2.
\end{aligned}$$

Since the last two terms in the right hand side are nonpositive, we deduce that

$$\tfrac{A_N}{1+\sigma}(h(x_N) - h_*) \leq \tfrac{1}{4}\|x_0 - x_*\|^2.$$

For the case $\sigma = 0$ [Güler, 1992, Theorem 6.1] provides a proof when using constant step sizes. Here, we follow the same pattern as before for allowing variable step sizes. We consider the following inequalities

- convexity between $x_k$ and $x_{k+1}$ with weight $\nu_{k,k+1} = A_k$
  (for $k = 0, \ldots, N-1$)

$$h(x_k) \geq h(x_{k+1}) + \langle g_{k+1}; x_k - x_{k+1} \rangle,$$

- convexity between $x_*$ and $x_k$ with weight $\nu_{*,k} = A_k - A_{k-1}$
  (for $k = 1, \ldots, N$)

$$h(x_*) \geq h(x_k) + \langle g_k; x_* - x_k \rangle.$$

As previously linearly combining the previous inequalities leads to

$$\begin{aligned}
\sum_{k=1}^{N-1} \nu_{k,k+1} h(x_k) + \sum_{k=1}^{N} \nu_{*,k} h(x_*) \geq & \sum_{k=1}^{N-1} \nu_{k,k+1} [h(x_{k+1}) + \langle g_{k+1}; x_k - x_{k+1} \rangle] \\
& + \sum_{k=1}^{N} \nu_{*,k} [h(x_k) + \langle g_k; x_* - x_k \rangle],
\end{aligned}$$

which can be reformulated exactly as

$$\begin{aligned} A_N(h(x_N) - h_*) \leq& \tfrac{1}{4}\|x_0 - x_*\|^2 - \tfrac{1}{4}\|x_* - x_0 + 2\sum_{i=1}^{N}(A_i - A_{i-1})g_i\|^2 \\ \leq& \tfrac{1}{4}\|x_0 - x_*\|^2. \end{aligned}$$

For the tightness part of the proof, we verify that the guarantee for (ORI-PPA) provided by Theorem 5.4.1 is actually non-improvable. That is, for all $\{\lambda_k\}_k$ with $\lambda_k > 0$, $\sigma \in [0, 1]$, $d \in \mathbb{N}$, $x_0 \in \mathbb{R}^d$, and $N \in \mathbb{N}$, there exists $f \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ such that this bound is achieved with equality. For proving this statement, it is sufficient to exhibit a one-dimensional function for which the bound is attained, which is what we do below. The bound is attained on the one-dimensional linear minimization problem

$$\min_x \{f(x) \equiv c\,x + i_{\mathbb{R}_+}(x)\}, \tag{5.18}$$

with an appropriate choice of $c > 0$, where $i_{\mathbb{R}_+}$ denotes the convex indicator function of $\mathbb{R}_+$. Indeed, one can check that the relative error criterion

$$\exists u_k \in \mathbb{R}_+,\ \tfrac{\lambda_k}{2}\|e_k\|^2 + f(x_k) - f(u_k) - \langle g_k; x_k - u_k\rangle \leq \tfrac{\lambda_k\sigma^2}{2}\|e_k + g_k\|^2$$

is satisfied with equality when picking $g_k = c$ ($g_k$ is thus a subgradient at $x_k$), $u_k = x_k$, and $e_k = -\frac{c\sigma}{1+\sigma}$; and hence $x_k = y_{k-1} - \frac{c\lambda_k}{1+\sigma}$. The argument is then as follows: if for some $x_0 > 0$ and $0 \leq h \leq x_0/c$ we manage to show that $x_N = x_0 - ch$, then $f(x_N) - f(x_*) = c(x_0 - ch)$ and hence the value of $c$ producing the worst possible (maximal) value of $f(x_N)$ is $c = \frac{x_0}{2h}$. In that case, the resulting value is $f(x_N) - f(x_*) = \frac{x_0^2}{4h}$. Therefore, in order to prove that the guarantee from Theorem 5.4.1 cannot be improved, we show that $x_N = x_0 - \frac{A_N}{1+\sigma}c$ on the linear problem (5.18). It is easy to show that $x_1 = x_0 - \frac{A_1}{1+\sigma}c$ using $A_1 = \lambda_1$. The argument follows by induction: assuming $x_k = x_0 - \frac{A_k}{1+\sigma}c$, one can compute

$$\begin{aligned} x_{k+1} =& \tfrac{\lambda_{k+1}}{A_{k+1}-A_k}\left(x_0 - \tfrac{2}{1+\sigma}\sum_{i=1}^{k}(A_i - A_{i-1})g_i\right) + \left(1 - \tfrac{\lambda_{k+1}}{A_{k+1}-A_k}\right)x_k \\ & - \lambda_{k+1}(g_{k+1} + e_{k+1}) \\ =& \tfrac{\lambda_{k+1}}{A_{k+1}-A_k}\left(x_0 - \tfrac{2c}{1+\sigma}A_k\right) + \left(1 - \tfrac{\lambda_{k+1}}{A_{k+1}-A_k}\right)\left(x_0 - \tfrac{A_k}{1+\sigma}c\right) - \lambda_{k+1}\tfrac{c}{1+\sigma} \\ =& x_0 - \tfrac{c}{1+\sigma}\tfrac{2\lambda_{k+1}A_k + (A_{k+1}-A_k)A_k - \lambda_{k+1}A_k + \lambda_{k+1}(A_{k+1}-A_k)}{A_{k+1}-A_k} \\ =& x_0 - \tfrac{c}{1+\sigma}\tfrac{(A_{k+1}-A_k)A_k + \lambda_{k+1}A_{k+1}}{A_{k+1}-A_k} \\ =& x_0 - \tfrac{c}{1+\sigma}A_{k+1}, \end{aligned}$$

where the second equality follows from simple substitutions, and the last equalities follow from basic algebra and $\lambda_{k+1}A_{k+1} = (A_{k+1} - A_k)^2$. The desired statement is proved by picking $c = \frac{(1+\sigma)x_0}{2A_N}$, reaching $f(x_N) - f(x_*) = \frac{(1+\sigma)x_0^2}{A_N}$.

■

A classical lower bound on the value of the sequence $\{A_k\}_k$ shows that the previous bound is a $O(N^{-2})$ when the $\lambda_k$ are lower bounded by some positive constant.

**Lemma 5.4.2** (Lemma 3.7 of Monteiro and Svaiter [2013]). *Given a sequence $\{\lambda_k\}_k$ with $\lambda_k \geq 0$. Let $A_0 = 0$ and $A_{k+1} = A_k + \frac{\lambda_{k+1}+\sqrt{4\lambda_{k+1}A_k+\lambda_{k+1}^2}}{2}$ defined recursively, then*

$$A_k \geq \frac{1}{4}\left(\sum_{i=1}^{k}\sqrt{\lambda_k}\right)^2 \quad \text{for } k \geq 1.$$

**Remark 5.4.3.** *We emphasize that there is no constraint on the relation between primal and dual points outputted by the process hidden behind "Obtain". In particular, primal-dual pairs of the form $(x_k, \partial h(x_k))$ or $(x_k, \frac{y_{k-1}-x_k}{\lambda_k})$ can be used.*

## 5.5 Dealing with strongly convex objectives

In this section we present how the methodology detailed in Section 5.3 can be extended to support strongly convex functions. We illustrate it on the simple relatively inexact proximal method studied in Section 5.3.5 applied to strongly convex objectives.

For adjusting the performance estimation approach to strongly convex problems, we only need minor modifications. According to Taylor et al. [2017c, Corollary 2], for $\mu > 0$ and a set $S \in \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}$

$$\begin{aligned}\exists h \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d) : f = h(x), \quad g \in \partial h(x) \quad \forall (x,g,f) \in S \\ \Leftrightarrow f' \geq f + \langle g; x'-x\rangle + \tfrac{\mu}{2}\|x-x'\|^2 \quad \forall (x,g,f),(x',g',f') \in S.\end{aligned} \tag{5.19}$$

In order to analyze inexact proximal minimization methods on strongly convex functions, we can simply follow Section 5.3 replacing the use of (5.6) by that of (5.19).

Let us illustrate that statement by instantiating the *inexact proximal minimization algorithm* for strongly convex objectives. We recall the form of the updates

$$\begin{aligned} w_{k+1} &= w_k - \lambda_{k+1}(v_{k+1} + e_{k+1}) \\ \|e_{k+1}\|^2 &\leq \tfrac{\sigma^2}{\lambda_{k+1}^2}\|w_{k+1} - w_k\|^2, \end{aligned} \tag{5.20}$$

with $\{\lambda_k\}_k$ a sequence of nonnegative step sizes, $v_{k+1} \in \partial h(w_{k+1})$, $\{e_k\}_k$ a sequence of errors and $\sigma \in [0,1]$.

For $\mu > 0$ we can study the following performance estimation problem for $\mu$-strongly convex objective functions $h$. In order to derive simpler worst-case guarantees, we use as slightly different initial condition compared with the previous section, which is $h(w_0) - h(w_*) \leq R^2$ with $R \in \mathbb{R}^*$.

$$\begin{aligned} \max_{\substack{d,h \\ w_*,w_0,\ldots,w_N \in \mathbb{R}^d \\ g_0,\ldots,g_N \in \mathbb{R}^d \\ e_0,\ldots,e_N \in \mathbb{R}^d}} \quad & h(w_N) - h(w_*) \\ \text{s.t. } & h \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d), \quad w_* \in \operatorname*{argmin}_x h(x) \\ & h(w_0) - h(w_*) \leq R^2 \\ & w_1,\ldots,w_N \text{ satisfying (5.20)} \\ & \|e_k\|^2 \leq \tfrac{\sigma^2}{\lambda_k^2}\|w_k - w_{k-1}\|^2 \quad k = 1,\ldots,N. \end{aligned} \tag{5.21}$$

Following similar developments as those of Section 5.3 and using interpolation conditions (5.19) we get the semidefinite reformulation

$$\begin{aligned}
\max_{G\succeq 0,\, H}\ & H(\mathbf{h}_N - \mathbf{h}_*) \\
\text{s.t. }\ 0 \ &\geq H(\mathbf{h}_i - \mathbf{h}_j) + \mathbf{v}_i^T G(\mathbf{w}_j - \mathbf{w}_i) \\
&\quad + \tfrac{\mu}{2}(\mathbf{w}_j - \mathbf{w}_i)^T G(\mathbf{w}_j - \mathbf{w}_i) \quad \forall i,j \in \{*, 0, \dots, N\} \\
R^2 &\geq H(\mathbf{h}_0 - \mathbf{h}_*) \\
0 \ &\geq \mathbf{e}_i^T G \mathbf{e}_i - \tfrac{\sigma^2}{\lambda_i^2}(\mathbf{w}_i - \mathbf{w}_{i-1})^T G(\mathbf{w}_i - \mathbf{w}_{i-1}) \quad \forall i \in \{1, \dots, N\}.
\end{aligned} \tag{5.22}$$

As before, we exhibit a dual feasible point, and the proof relies on weak duality.

**Theorem 5.5.1.** *Let $\mu \geq 0$, $h \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$, a sequence of step sizes $\{\lambda_k\}_k$ with $\lambda_k > 0$, and $\sigma \in [0,1]$. For any starting point $w_0 \in \mathbb{R}^d$, $N \geq 1$, the iterates of (5.20) satisfy*

$$h(w_N) - h(w_*) \leq \prod_{i=1}^{N} \left(\tfrac{1+\sigma}{1+\sigma+\lambda_k\mu}\right)^2 (h(w_0) - h(w_*)),$$

*with $w_* \in \operatorname{argmin}_x h(x)$. Furthermore, this bound is tight: for all $\mu \geq 0$, $\{\lambda_k\}_k$ with $\lambda_k \geq 0$, $\sigma \in [0,1]$, $d \in \mathbb{N}$, $w_0 \in \mathbb{R}^d$, and $N \in \mathbb{N}$, there exists $h \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$ such that this bound is achieved with equality.*

*Proof.* Let us denote by $\rho_k = \frac{1+\sigma}{1+\sigma+\lambda_k\mu} \in [0,1]$ for $k = 1, \dots, N$. We show the result by proving that

$$h(w_k) - h(w_*) \leq \rho_k^2 (h(w_{k-1}) - h(w_*)) \quad k = 1 \dots, N.$$

Indeed, chaining these inequalities for $k \in [1, N]$ leads to the desired conclusion.

We first detail the case $\sigma \in (0,1]$. Let $k \in [1, N]$, and consider the following inequalities with their corresponding weights :

- strong convexity between $w_{k-1}$ and $w_k$ with weight $\nu_{k-1,k} = \rho_k^2$

$$h(w_{k-1}) \geq h(w_k) + \langle v_k; w_{k-1} - w_k \rangle + \tfrac{\mu}{2}\|w_{k-1} - w_k\|^2,$$

- strong convexity between $w_*$ and $w_k$ with weight $\nu_{*,k} = 1 - \rho_k^2$

$$h(w_*) \geq h(w_k) + \langle v_k; w_* - w_k \rangle + \tfrac{\mu}{2}\|w_k - w_*\|^2,$$

- approximation requirement on $w_k$ with weight $\nu_k = \frac{\lambda_k \rho_k}{2\sigma}$

$$\tfrac{\sigma^2}{\lambda_k^2}\|w_{k-1} - w_k\|^2 \geq \|e_k\|^2.$$

By linearly combining previous inequalities, with their corresponding weights (which are nonnegative), we arrive to the following valid inequality:

$$\begin{aligned}
&\nu_{k-1,k} h(w_{k-1}) + \nu_{*,k} h(w_*) + \nu_k \tfrac{\sigma^2}{\lambda_k^2}\|w_{k-1} - w_k\|^2 \\
&\geq \nu_{k-1,k}[h(w_k) + \langle v_k; w_{k-1} - w_k \rangle + \tfrac{\mu}{2}\|w_{k-1} - w_k\|^2] \\
&\quad + \nu_{*,k}[h(w_k) + \langle v_k; w_* - w_k \rangle + \tfrac{\mu}{2}\|w_* - w_k\|^2] + \nu_k \|e_k\|^2.
\end{aligned} \tag{5.23}$$

First we can regroup the function values together and observe that

$$\begin{aligned}&\nu_{k-1,k}h(w_k)+\nu_{*,k}h(w_k)-\nu_{k-1,k}h(w_{k-1})-\nu_{*,k}h(w_*)\\&=h(w_k)-h(w_*)-\rho_k^2(h(w_{k-1})-h(w_*)).\end{aligned}$$

Then, we regroup the vector variables together and use $w_k = w_{k-1} - \lambda_k(v_k + e_k)$ in

$$\begin{aligned}&\nu_{k-1,k}[\langle v_k; w_{k-1}-w_k\rangle+\tfrac{\mu}{2}\|w_{k-1}-w_k\|^2]+\nu_k[\|e_k\|^2-\tfrac{\sigma^2}{\lambda_k^2}\|w_{k-1}-w_k\|^2]\\&\quad+\nu_{*,k}[\langle v_k; w_*-w_k\rangle+\tfrac{\mu}{2}\|w_*-w_k\|^2]\\&=\nu_{k-1,k}[\lambda_k\langle v_k; v_k+e_k\rangle+\tfrac{\mu\lambda_k^2}{2}\|v_k+e_k\|^2]+\nu_k[\|e_k\|^2-\sigma^2\|v_k+e_k\|^2]\\&\quad+\nu_{*,k}[\langle v_k; w_*-w_{k-1}+\lambda_k(v_k+e_k)\rangle+\tfrac{\mu}{2}\|w_*-w_{k-1}+\lambda_k(v_k+e_k)\|^2]\\&=\nu_{k-1,k}[\lambda_k\langle v_k; v_k+e_k\rangle+\tfrac{\mu\lambda_k^2}{2}\|v_k+e_k\|^2]+\nu_k[\|e_k\|^2-\sigma^2\|v_k+e_k\|^2]\\&\quad+\nu_{*,k}[\tfrac{\mu}{2}\|w_*-w_{k-1}+\lambda_k(v_k+e_k)+\tfrac{1}{\mu}v_k\|^2-\tfrac{1}{2\mu}\|v_k\|^2].\end{aligned}$$

We can then factorize the following expression

$$\begin{aligned}&\nu_{k-1,k}[\lambda_k\langle v_k; v_k+e_k\rangle+\tfrac{\mu\lambda_k^2}{2}\|v_k+e_k\|^2]+\nu_k[\|e_k\|^2-\sigma^2\|v_k+e_k\|^2]-\tfrac{\nu_{*,k}}{2\mu}\|v_k\|^2\\&=[\nu_{k-1,k}\tfrac{\mu\lambda_k^2}{2}+\nu_k(1-\sigma^2)]\|e_k\|^2+[\nu_{k-1,k}\lambda_k+\nu_{k-1,k}\mu\lambda_k^2-2\nu_k\sigma^2]\langle e_k; v_k\rangle\\&\quad+[\nu_{k-1,k}\lambda_k+\nu_{k-1,k}\tfrac{\mu\lambda_k^2}{2}-\nu_k\sigma^2-\tfrac{\nu_{*,k}}{2\mu}]\|v_k\|^2\\&=\tfrac{\lambda_k\rho_k}{2\sigma}[\lambda_k\mu\sigma\rho_k+(1-\sigma^2)]\|e_k\|^2+\lambda_k\rho_k[\rho_k(1+\lambda_k\mu)-\sigma]\langle e_k; v_k\rangle\\&\quad+[\tfrac{(\rho_k+\lambda_k\mu\rho_k)^2-(1+\lambda_k\mu\sigma\rho_k)}{2\mu}]\|v_k\|^2\\&=\tfrac{\lambda_k(1+\sigma)^2(1-\sigma^2+\lambda_k\mu)}{2\sigma(1+\sigma+\lambda_k\mu)^2}\|e_k\|^2+\tfrac{\lambda_k(1+\sigma)(1-\sigma^2+\lambda_k\mu)}{(1+\sigma+\lambda_k\mu)^2}\langle e_k; v_k\rangle+\tfrac{\lambda_k\sigma(1-\sigma^2+\lambda_k\mu)}{2(1+\sigma+\lambda_k\mu)^2}\|v_k\|^2\\&=\tfrac{\lambda_k(1+\sigma)^2(1-\sigma^2+\lambda_k\mu)}{2\sigma(1+\sigma+\lambda_k\mu)^2}\|e_k+\tfrac{\sigma}{1+\sigma}v_k\|^2,\end{aligned}$$

where we replaced $\rho_k$ by its expression in the second to last line. Finally (5.23) can be written as

$$\begin{aligned}0\geq\,&h(w_k)-h(w_*)-\rho_k^2(h(w_{k-1})-h(w_*))\\&+\tfrac{(1-\rho_k^2)\mu}{2}\|w_*-w_{k-1}+\lambda_k(v_k+e_k)+\tfrac{1}{\mu}v_k\|^2\\&+\tfrac{\lambda_k(1+\sigma)^2(1-\sigma^2+\lambda_k\mu)}{2\sigma(1+\sigma+\lambda_k\mu)^2}\|e_k+\tfrac{\sigma}{1+\sigma}v_k\|^2.\end{aligned}$$

Since $\rho_k, \sigma \in [0, 1]$ the leading factors in front of the squared Euclidean norms are nonnegative and this leads to

$$h(w_k) - h(w_*) \leq \rho_k^2(h(w_{k-1}) - h(w_*))$$

which concludes the first part of the proof for $\sigma \in (0, 1]$, according to our initial remark.

For the exact case (i.e., $\sigma = 0$), the proof carries on likewise, by only combining the first two inequalities, encoding strong convexity, leading to

$$0\geq h(w_k)-h(w_*)-\rho_k^2(h(w_{k-1})-h(w_*))+\tfrac{(1-\rho_k^2)\mu}{2}\|w_*-w_{k-1}+\left(\lambda_k+\tfrac{1}{\mu}\right)v_k\|^2,$$

and the desired conclusion follows. For tightness part of the proof, we show that the guarantee provided in Theorem 5.5.1 is non-improvable. That is, for all $\mu \geq 0$, $\{\lambda_k\}_k$ with $\lambda_k \geq 0$, $\sigma \in [0, 1]$, $d \in \mathbb{N}$,

$w_0 \in \mathbb{R}^d$, and $N \in \mathbb{N}^*$, there exists $h \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$ such that this bound is achieved with equality. Indeed, the bound is attained on the simple quadratic minimization problem

$$\min_x \{h(x) \equiv \tfrac{\mu}{2}\|x\|^2\}. \tag{5.24}$$

We can check that the relative error criterion

$$\tfrac{\lambda_k}{2}\|e_k\|^2 \leq \tfrac{\sigma^2\lambda_k}{2}\|e_k + v_k\|^2,$$

is satisfied with equality when picking $v_k = \nabla h(w_{k+1}) = \mu w_{k+1}$ and $e_k = -\frac{\sigma}{1+\sigma}v_k$. Under these choices, one can write

$$w_{k+1} = w_k - \tfrac{\lambda_{k+1}\mu}{1+\sigma}w_{k+1},$$

which leads to

$$w_{k+1} = \tfrac{1+\sigma}{1+\sigma+\lambda_{k+1}}w_k.$$

Finally

$$w_N = \prod_{i=1}^{N} \tfrac{1+\sigma}{1+\sigma+\lambda_i\mu}w_0,$$

and the desired results follows. ■

## 5.6 Conclusion

In this chapter, we showed that the performance estimation framework, initiated by Drori and Teboulle [2014], allows studying first-order methods involving natural notions of inexact proximal operations. On the way, we reviewed natural accuracy requirements used in the literature for characterizing inexact proximal operations. We also used the approach for optimizing the parameters of an inexact proximal point algorithm. Finally, we presented a simple extension to the strongly convex setting.

As future works, we believe the approach can be extended to inexact Bregman proximal steps (see e.g., Eckstein [1998]), and to inexact resolvent for monotone operators (see e.g., Solodov and Svaiter [1999a]), for example by following steps taken Dragomir et al. [2021], Ryu et al. [2020]. Further using those tools for designing optimized methods involving inexact proximal operations for monotone inclusions, and variational inequalities are also possibilities. Let us also mention that it is currently unclear to us whether similar techniques can be used for studying higher-order proximal methods, as recently introduced by Nesterov [Nesterov, 2020b,a].

Finally, an alternate, and more geometric, approach for studying inexact proximal operations could be to extend *scaled relative graphs* Ryu et al. [2019] to deal with inaccuracies.

**Codes** Codes, that include notebooks for helping the reader reproducing the proofs and implementation of the performance estimation problems, are available at

`https://github.com/mathbarre/InexactProximalOperators/tree/version-2`

Notions of inexactness were also included in the performance estimation toolbox from Taylor et al. [2017b].

# Appendices

## 5.A More examples of fixed-step inexact proximal methods

This extends the list of examples of Section 5.3.1.

- The *hybrid approximate extragradient algorithm* (see Solodov and Svaiter [1999a] or Monteiro and Svaiter [2010, Section 4]) can be described as

$$x_{k+1} = x_k - \eta_{k+1} g_{k+1},$$

such that $\exists u_{k+1}, \mathrm{PD}_{\eta_{k+1}h}(u_{k+1}, g_{k+1}; x_k) \leq \frac{\sigma^2}{2}\|u_{k+1} - x_k\|^2$ (see Lemma 5.2.1 for a link between $\varepsilon$-subgradient formulation and primal-dual gap). One iteration of this form can be artificially cast into three iterations of (5.1) as

$$\left\{\begin{array}{rcl} w_{3k+1} & = & w_{3k} - e_{3k} \\ w_{3k+2} & = & w_{3k+1} - e_{3k+1} \\ w_{3k+3} & = & w_{3k+2} + e_{3k} + e_{3k+1} - \eta_{k+1} v_{3k+2} \end{array}\right.$$

with $v_{3k+2} \in \partial h(w_{3k+2})$ . This corresponds to setting $\lambda_{3k+1} = \lambda_{3k+2} = \lambda_{3k+3} = 0$, $\alpha_{3k+3,3k+2} = \eta_{k+1}$, $\beta_{3k+1,3k} = \beta_{3k+2,3k+1} = 1$, $\beta_{3k+3,3k+1} = \beta_{3k+3,3k+2} = -1$ and the other parameters to zero. Notice that $w_{3k+3} = w_{3k} - \eta_{k+1} v_{3k+2}$ and imposing the constraint $\mathrm{PD}_{\eta_{k+1}h}(w_{3k+1}, v_{3k+2}; w_{3k}) \leq \frac{\sigma^2}{2}\|w_{3k+1} - w_{3k}\|^2$ allows identifying the primal-dual pair $(u_{k+1}, g_{k+1})$ with $(w_{3k+1}, v_{3k+2})$ and iterates $x_{k+1}$ with $w_{3k+3}$. In addition, we set

$$\begin{aligned} \mathrm{EQ}_{3k+1} &= 0, \\ \mathrm{EQ}_{3k+2} &= 0, \\ \mathrm{EQ}_{3k+3} &= \mathrm{PD}_{\eta_{k+1}h}(w_{3k+1}, v_{3k+2}; w_{3k}) - \tfrac{\sigma^2}{2}\|w_{3k+1} - w_{3k}\|^2. \end{aligned}$$

Using $v_{3k+2} \in \partial h(w_{3k+2})$, we have $h^*(v_{3k+2}) = \langle v_{3k+2}; w_{3k+2}\rangle - h(w_{3k+2})$ and thus

$$\begin{aligned} \mathrm{EQ}_{3k+3} = & \tfrac{1}{2}\|w_{3k+1} - w_{3k+3}\|^2 + \eta_{k+1}(h(w_{3k+1}) - h(w_{3k+2}) \\ & - \langle v_{3k+2}; w_{3k+1} - w_{3k+2}\rangle) - \tfrac{\sigma^2}{2}\|w_{3k+1} - w_{3k}\|^2, \end{aligned}$$

which complies with (5.2) and is Gram-representable.

- The *inexact accelerated proximal point algorithm IAPPA1* in its form from Salzo and Villa [2012, Section 5] can be written as

$$\left\{\begin{array}{rcl} t_{k+1} & = & \frac{1+\sqrt{1+4t_k^2\frac{\eta_{k+1}}{\eta_{k+2}}}}{2} \\ x_{k+1} & = & y_k - \eta_{k+1}(g_{k+1} + r_{k+1}) \\ y_{k+1} & = & x_{k+1} + \frac{t_k - 1}{t_{k+1}}(x_{k+1} - x_k) \end{array}\right.$$

with $t_0 = 1$, $\{\eta_k\}_k$ a sequence of step sizes, $y_0 = x_0 \in \mathbb{R}^d$ along with an inexactness criterion of the form $\mathrm{PD}_{\eta_{k+1}h}(x_{k+1}, g_{k+1}; y_k) \leq \varepsilon_{k+1}$ given a nonnegative sequence $\{\varepsilon_k\}_k$. Similarly to Güler's method we get the recursive formulation

$$x_{k+2} = \left(1 + \tfrac{t_k - 1}{t_{k+1}}\right) x_{k+1} - \tfrac{t_k - 1}{t_{k+1}} x_k - \eta_{k+2}(g_{k+2} + r_{k+2}).$$

We consider particular iterations from (5.1) of the form

$$\begin{cases} w_{2k+1} &= w_{2k} - e_{2k} \\ w_{2k+2} &= w_{2k+1} - \sum_{i=1}^{2k+1} \alpha_{2k+2,i} v_i - \sum_{i=0}^{2k+1} \beta_{2k+2,i} e_i, \end{cases}$$

with initial iterate $w_0 = x_0$. We aim at finding parameters $\alpha_{i,j}$, $\beta_{i,j}$ such that we can identify $\{w_{2k}\}_k$ with $\{x_k\}_k$ (i.e., any sequence $\{x_k\}_k$ can be obtained as a sequence $\{w_{2k}\}_k$). We set $\alpha_{2k+2,2k+1} = \beta_{2k+2,2k+1} = \eta_{k+1}$, $\alpha_{2k+2,i} = \frac{t_{k-1}-1}{t_k}\alpha_{2k,i}$ for $i = 1, \dots, 2k-1$ and $\beta_{2k+2,i} = \frac{t_{k-1}-1}{t_k}\beta_{2k,i}$ for $i \in \{0, \dots, 2k-1\} \backslash \{2(k-1)\}$ as well as $\beta_{2k+2,2k} = -1$ and $\beta_{2k+2,2(k-1)} = \frac{t_{k-1}-1}{t_k}(1 + \beta_{2k,2(k-1)})$.
This gives

$$\begin{aligned} w_{2(k+1)} =& w_{2k+1} + e_{2k} - \tfrac{t_{k-1}-1}{t_k}(e_{2(k-1)}) - \tfrac{t_{k-1}-1}{t_k} \sum_{i=1}^{2k-1} \alpha_{2k,i} v_i \\ & - \tfrac{t_{k-1}-1}{t_k} \sum_{i=0}^{2k-1} \beta_{2k,i} e_i - \eta_{k+1}(v_{2k+1} + e_{2k+1}) \\ =& (1 + \tfrac{t_{k-1}-1}{t_k}) w_{2k} - \tfrac{t_{k-1}-1}{t_k} w_{2(k-1)} - \eta_{k+1}(v_{2k+1} + e_{2k+1}), \end{aligned}$$

which shows that $\{w_{2k}\}_k$ follows the same recursive equation as $\{x_k\}_k$. In addition, we have $w_0 = x_0$ and $w_2 = x_0 - \eta_1(v_1 + e_1)$ similar to $x_1 = x_0 - \eta_1(g_1 + r_1)$.
Requiring $\mathrm{PD}_{\eta_{k+1}h}(w_{2k+2}, v_{2k+1};\ w_{2k+2} + \eta_{k+1}(v_{2k+1} + e_{2k+1})) \leq \varepsilon_{k+1}$ (with the convention $w_{-1} = w_0$) allows to identify the primal-dual pair $(x_{k+1}, g_{k+1})$ with $(w_{2k+2}, v_{2k+1})$.

Finally, we can set $\mathrm{EQ}_{2k+2} = \mathrm{PD}_{\eta_{k+1}h}(w_{2k+2}, v_{2k+1};\ w_{2k+2} + \eta_{k+1}(v_{2k+1} + e_{2k+1})) - \varepsilon_{k+1}$ which is Gram-representable (similar to *hybrid approximate extragradient algorithm*).

Note that we can proceed similarly for *IAPPA2* from Salzo and Villa [2012, Section 5] with sequence $\{a_k\}_k$ constant equal to 1, by removing the sequence $\{r_k\}_k$ "type 2" errors).

- The *acceleration hybrid proximal extragradient algorithm* (A-HPE) from Monteiro and Svaiter [2013, Section 3] can be written as

$$\begin{cases} a_{k+1} &= \frac{\eta_{k+1} + \sqrt{\eta_{k+1}^2 + 4\eta_{k+1} A_k}}{2} \\ A_{k+1} &= A_k + a_{k+1} \\ \tilde{x}_k &= y_k + \frac{a_{k+1}}{A_{k+1}}(x_k - y_k) \\ y_{k+1} &= \tilde{x}_k - \eta_{k+1}(g_{k+1} + r_{k+1}) \\ x_{k+1} &= x_k - a_{k+1} g_{k+1}, \end{cases}$$

with $A_0 = 0$, $\{\eta_k\}_k$ a sequence of step sizes, $y_0 = x_0 \in \mathbb{R}^d$ along with an inexactness criterion of the form $\mathrm{PD}_{\eta_{k+1}h}(y_{k+1}, g_{k+1}; \tilde{x}_k) \leq \frac{\sigma}{2}\|y_{k+1} - \tilde{x}_k\|^2$ given a parameter $\sigma \in [0, 1]$. As in the previous examples, we search for a recursive equation followed by the sequence $\{y_k\}_k$. By performing multiple substitutions, we obtain

$$\begin{aligned} y_{k+2} =& \tilde{x}_{k+1} - \eta_{k+2}(g_{k+2} + r_{k+2}) \\ =& \tfrac{A_{k+1}}{A_{k+2}} y_{k+1} + \tfrac{a_{k+2}}{A_{k+2}} x_{k+1} - \eta_{k+2}(g_{k+2} + r_{k+2}) \end{aligned}$$

$$
\begin{aligned}
=&\tfrac{A_{k+1}}{A_{k+2}}y_{k+1} + \tfrac{a_{k+2}}{A_{k+2}}\left(x_k - a_{k+1}g_{k+1}\right) - \eta_{k+2}(g_{k+2}+r_{k+2})\\
=&\tfrac{A_{k+1}}{A_{k+2}}y_{k+1} + \tfrac{a_{k+2}}{A_{k+2}}\left(\tfrac{A_{k+1}}{a_{k+1}}\tilde{x}_k - \tfrac{A_k}{a_{k+1}}y_k - a_{k+1}g_{k+1}\right) - \eta_{k+2}(g_{k+2}+r_{k+2})\\
=&\tfrac{A_{k+1}}{A_{k+2}}y_{k+1} + \tfrac{a_{k+2}}{A_{k+2}}\left(\tfrac{A_{k+1}}{a_{k+1}}\left(y_{k+1}+\eta_{k+1}(g_{k+1}+r_{k+1})\right) - \tfrac{A_k}{a_{k+1}}y_k - a_{k+1}g_{k+1}\right)\\
&- \eta_{k+2}(g_{k+2}+r_{k+2})\\
=&\left(\tfrac{A_{k+1}}{A_{k+2}} + \tfrac{a_{k+2}A_{k+1}}{A_{k+2}a_{k+1}}\right)y_{k+1} - \tfrac{a_{k+2}A_k}{A_{k+2}a_{k+1}}y_k + \tfrac{a_{k+2}}{A_{k+2}}\left(\tfrac{A_{k+1}}{a_{k+1}}\eta_{k+1} - a_{k+1}\right)g_{k+1}\\
&+ \tfrac{a_{k+2}A_{k+1}}{A_{k+2}a_{k+1}}\eta_{k+1}r_{k+1} - \eta_{k+2}(g_{k+2}+r_{k+2})\\
=&\left(1 + \tfrac{a_{k+2}A_k}{A_{k+2}a_{k+1}}\right)y_{k+1} - \tfrac{a_{k+2}A_k}{A_{k+2}a_{k+1}}y_k + \tfrac{a_{k+2}}{A_{k+2}}\left(\tfrac{A_{k+1}}{a_{k+1}}\eta_{k+1} - a_{k+1}\right)g_{k+1}\\
&+ \tfrac{a_{k+2}A_{k+1}}{A_{k+2}a_{k+1}}\eta_{k+1}r_{k+1} - \eta_{k+2}(g_{k+2}+r_{k+2}).
\end{aligned}
$$

Similar to *IAPPA1*, we consider particular iterations from (5.1) of the form

$$
\begin{cases}
w_{2k+1} &= w_{2k} - e_{2k}\\
w_{2k+2} &= w_{2k+1} - \sum_{i=1}^{2k+1}\alpha_{2k+2,i}v_i - \sum_{i=0}^{2k+1}\beta_{2k+2,i}e_i,
\end{cases}
$$

with initial iterate $w_0 = x_0$. We aim at finding parameters $\alpha_{i,j}$, $\beta_{i,j}$ such that we can identify $\{w_{2k}\}_k$ with $\{y_k\}_k$ (i.e., any sequence $\{y_k\}_k$ can be obtained as a sequence $\{w_{2k}\}_k$). We set $\alpha_{2(k+1),2k+1} = \beta_{2(k+1),2k+1} = \eta_{k+1}$, $\alpha_{2(k+1),i} = \frac{a_{k+1}A_{k-1}}{A_{k+1}a_k}\alpha_{2k,i}$ for $i \in \{1,\dots,2(k-1)\}$ and $\beta_{2(k+1),i} = \frac{a_{k+1}A_{k-1}}{A_{k+1}a_k}\beta_{2k,i}$ for $i \in \{0,\dots,2k-3\}$ as well as $\beta_{2(k+1),2k} = -1$, $\beta_{2(k+1),2k-1} = \frac{a_{k+1}}{A_{k+1}a_k}(A_{k-1}\beta_{2k,2k-1} - A_k\eta_k)$, $\beta_{2(k+1),2(k-1)} = \frac{a_{k+1}A_{k-1}}{A_{k+1}a_k}(1+\beta_{2k,2(k-1)})$ and $\alpha_{2(k+1),2k-1} = \frac{a_{k+1}}{A_{k+1}}\left(\frac{A_{k-1}}{a_k}\alpha_{2k,2k-1} - \frac{A_k}{a_k}\eta_k + a_k\right)$.

This gives

$$
\begin{aligned}
w_{2(k+1)} =&w_{2k+1} + e_{2k} + \tfrac{a_{k+1}A_k}{A_{k+1}a_k}\eta_k e_{2k-1} - \tfrac{a_{k+1}A_{k-1}}{A_{k+1}a_k}e_{2(k-1)} + \tfrac{a_{k+1}}{A_{k+1}}\left(\tfrac{A_k}{a_k}\eta_k - a_k\right)v_{2k-1}\\
&- \tfrac{a_{k+1}A_{k-1}}{A_{k+1}a_k}\sum_{i=1}^{2k-1}\alpha_{2k,i}v_i - \tfrac{a_{k+1}A_{k-1}}{A_{k+1}a_k}\sum_{i=0}^{2k-1}\beta_{2k,i}e_i - \eta_{n+1}(v_{2k+1}+e_{2k+1})\\
=&w_{2k} + \tfrac{a_{k+1}A_k}{A_{k+1}a_k}\eta_k e_{2k-1} - \tfrac{a_{k+1}A_{k-1}}{A_{k+1}a_k}e_{2(k-1)} + \tfrac{a_{k+1}}{A_{k+1}}\left(\tfrac{A_k}{a_k}\eta_k - a_k\right)v_{2k-1}\\
&+ \tfrac{a_{k+1}A_{k-1}}{A_{k+1}a_k}(w_{2k} - w_{2(k-1)} + e_{2(k-1)}) - \eta_{n+1}(v_{2k+1}+e_{2k+1})\\
=&\left(1 + \tfrac{a_{k+1}A_{k-1}}{A_{k+1}a_k}\right)w_{2k} - \tfrac{a_{k+1}A_{k-1}}{A_{k+1}a_k}w_{2(k-1)} + \tfrac{a_{k+1}}{A_{k+1}}\left(\tfrac{A_k}{a_k}\eta_k - a_k\right)v_{2k-1}\\
&+ \tfrac{a_{k+1}A_k}{A_{k+1}a_k}\eta_k e_{2k-1} - \eta_{n+1}(v_{2k+1}+e_{2k+1}),
\end{aligned}
$$

which shows that $\{w_{2k}\}_k$ follows the same recursive equation as $\{y_k\}_k$. In addition, we have $w_0 = x_0 = y_0$ and $w_2 = y_0 - \eta_1(v_1+e_1)$ similar to $x_1 = x_0 - \eta_1(g_1+r_1)$. Requiring $\mathrm{PD}_{\eta_{k+1}h}(w_{2(k+1)}, v_{2k+1};\, w_{2k+2} + \eta_{k+1}(v_{2k+1}+e_{2k+1})) \le \frac{\sigma^2}{2}\|w_{2(k+1)} - w_{2k}\|^2$ allows to identify the primal-dual pair $(y_{k+1}, g_{k+1})$ with $(w_{2(k+1)}, v_{2k+1})$.

Finally, we set

$$
\mathrm{EQ}_{2k+2} = \mathrm{PD}_{\eta_{k+1}h}(w_{2k+2}, v_{2k+1};\, w_{2k+2} + \eta_{k+1}(v_{2k+1}+e_{2k+1})) - \tfrac{\sigma^2}{2}\|w_{2(k+1)} - w_{2k}\|^2,
$$

which is Gram-representable (similar to *hybrid approximate extragradient algorithm*).

## 5.B Interpolation with $\varepsilon$-subdifferentials

In this section, we provide the interpolation result for working with $\varepsilon$-subdifferentials inside performance estimation problems.

**Theorem 5.B.1.** *Let $I$ be a finite set of indices and $S = \{(w_i, v_i, h_i, \varepsilon_i)\}_{i\in I}$ with $w_i, v_i \in \mathbb{R}^d$, $h_i, \varepsilon_i \in \mathbb{R}$ for all $i \in I$. There exists $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ satisfying*

$$h_i = h(w_i), \text{ and } v_i \in \partial_{\varepsilon_i} h(w_i) \text{ for all } i \in I \tag{5.25}$$

*if and only if*

$$h_i \geq h_j + \langle v_j; w_i - w_j\rangle - \varepsilon_j \tag{5.26}$$

*holds for all $i, j \in I$.*

*Proof.* ($\Rightarrow$) Assuming $h \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ and (5.25), the inequalities (5.26) hold by definition.

($\Leftarrow$) Assuming (5.26) hold, one can perform the following construction:

$$\tilde{h}(x) = \max_i \{h_i + \langle v_i; x - w_i\rangle - \varepsilon_i\},$$

and one can easily check that $h = \tilde{h} \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ satisfies (5.25). ■

## 5.C Equivalence with Güler's method

In this section, we show that optimized algorithm (ORI-PPA) and Güler's second method [Güler, 1992, Section 6] are equivalent (i.e., produce the same iterates), in the case of exact proximal computations (i.e., $\sigma = 0$).

We consider a constant sequence of step sizes $\{\lambda_k\}_k$ with $\lambda_k = \lambda > 0$. In Güler's second method, the sequence $\{\beta_k\}_k$ is defined as $\beta_1 = 1$ and

$$\beta_{k+1} = \tfrac{1+\sqrt{4\beta_k^2+1}}{2}.$$

The sequence $\{A_k\}_k$ generated by (ORI-PPA) satisfies $A_0 = 0$ and

$$A_{k+1} = A_k + \tfrac{\lambda+\sqrt{4\lambda A_k+\lambda^2}}{2}, \quad k \geq 0.$$

We can link together these two sequences through the following equality

$$\beta_k = \tfrac{A_k - A_{k-1}}{\lambda}, \quad k \geq 1. \tag{5.27}$$

Let us prove it recursively. First, observe that $\beta_1 = 1$ and $\frac{A_1-A_0}{\lambda} = 1$. Then assuming that the property is true for some $k \geq 1$, we have

$$\begin{aligned}\beta_{k+1} &= \tfrac{1+\sqrt{4\beta_k^2+1}}{2} \\ &= \tfrac{1+\sqrt{4\frac{(A_{k+1}-A_k)^2}{\lambda^2}+1}}{2}.\end{aligned}$$

One might notice that

$$(A_{k+1} - A_k)^2 = \tfrac{2\lambda^2+4\lambda A_k+2\lambda\sqrt{4\lambda A_k+\lambda^2}}{4} = \lambda A_{k+1},$$

which gives

$$\begin{aligned}\beta_{k+1} &= \frac{1+\sqrt{4\frac{A_{k+1}}{\lambda}+1}}{2} \\ &= \frac{\lambda+\sqrt{4\lambda A_{k+1}+\lambda^2}}{2\lambda} \\ &= \frac{A_{k+2}-A_{k+1}}{\lambda},\end{aligned}$$

and we finally arrive to (5.27).

In the exact case ($\sigma = 0$) iterations of (ORI-PPA) can be written as

$$\left\{\begin{array}{rcl} y_k &=& x_k + \frac{\lambda}{A_{k+1}-A_k}(z_k - x_k) \\ x_{k+1} &=& \mathrm{prox}_{\lambda h}(y_k) \\ z_{k+1} &=& z_k + \frac{2(A_{k+1}-A_k)}{\lambda}(x_{k+1} - y_k). \end{array}\right.$$

Therefore, we can express

$$\begin{aligned} y_{k+1} &= x_{k+1} + \tfrac{\lambda}{A_{k+2}-A_{k+1}}(z_k + \tfrac{2(A_{k+1}-A_k)}{\lambda}(x_{k+1} - y_k) - x_{k+1}) \\ &= x_{k+1} + \tfrac{\lambda}{A_{k+2}-A_{k+1}}(x_k - \tfrac{A_{k+1}-A_k}{\lambda}(x_k - y_k) + \tfrac{2(A_{k+1}-A_k)}{\lambda}(x_{k+1} - y_k) - x_{k+1}) \\ &= x_{k+1} + \tfrac{\lambda}{A_{k+2}-A_{k+1}}\left(\left(\tfrac{A_{k+1}-A_k}{\lambda} - 1\right)(x_{k+1} - x_k) + \tfrac{A_{k+1}-A_k}{\lambda}(x_{k+1} - y_k)\right), \end{aligned}$$

and combining the last equality with (5.27) leads to

$$y_{k+1} = x_{k+1} + \tfrac{\beta_{k+1}-1}{\beta_{k+2}}(x_{k+1} - x_k) + \tfrac{\beta_{k+1}}{\beta_{k+2}}(x_{k+1} - y_k)$$

which is exactly the update in Güler's second method [Güler, 1992, Section 6] modulo a translation in the indices of the $\{y_k\}_k$ sequence (indeed in Güler's method $y_1 = x_0$ whereas in (ORI-PPA) $y_0 = x_0$).

## 5.D Missing details in Theorem 5.4.1

The missing elements in the proof of Theorem 5.4.1 are presented bellow.

*Proof.* Let us rewrite the method in terms of a single sequence, by substitution of $y_k$ and $z_k$:

$$\begin{aligned} e_k &:= \tfrac{1}{\lambda_k}(y_{k-1} - \lambda_k g_k - x_k) \\ x_k &= \tfrac{\lambda_k}{A_k - A_{k-1}}\left(x_0 - \tfrac{2}{1+\sigma}\sum_{i=1}^{k-1}(A_i - A_{i-1})g_i\right) + \left(1 - \tfrac{\lambda_k}{A_k - A_{k-1}}\right)x_{k-1} - \lambda_k(g_k + e_k), \end{aligned} \tag{5.28}$$

and let us state the following identity on the $A_k$ coefficients

$$\lambda_{k+1}A_{k+1} = (A_{k+1} - A_k)^2 \text{ (for } k \geq 0). \tag{5.29}$$

We prove the desired convergence result by induction. First, for $N = 1$

$$\begin{aligned} 0 \geq& \nu_{*,1}[h(u_1) - h(x_*) + \langle g_1; x_* - u_1\rangle] + \nu_{1,1}[h(u_1) - h(x_1) + \langle g_1; x_1 - u_1\rangle] \\ &+ \nu_1[\tfrac{\lambda_1}{2}\|e_1\|^2 - \tfrac{\lambda_1\sigma^2}{2}\|e_1 + g_1\|^2 + h(x_1) - h(u_1) - \langle g_1; x_1 - u_1\rangle] \end{aligned}$$

with $\nu_{*,1} = \frac{A_1 - A_0}{1+\sigma} = \frac{A_1}{1+\sigma}$ as $A_0 = 0$, $\nu_{1,1} = \frac{(1-\sigma)A_1}{\sigma(1+\sigma)}$ and $\nu_1 = \frac{A_1}{\sigma(1+\sigma)}$. This gives

$$
\begin{aligned}
0 \geq & \tfrac{A_1}{1+\sigma}(h(x_1) - h_*) + \tfrac{A_1}{1+\sigma}\langle g_1; x_* - x_1\rangle + \tfrac{A_1}{\sigma(1+\sigma)}[\tfrac{1}{2}\|e_1\|^2 - \tfrac{\sigma^2}{2}\|e_1 + g_1\|^2] \\
= & \tfrac{A_1}{1+\sigma}(h(x_1) - h_*) + \tfrac{A_1}{1+\sigma}\langle g_1; x_* - x_0 + \lambda_1(g_1 + e_1)\rangle + \tfrac{A_1}{\sigma(1+\sigma)}[\tfrac{\lambda_1}{2}\|e_1\|^2 - \tfrac{\lambda_1\sigma^2}{2}\|e_1 + g_1\|^2] \\
= & \tfrac{A_1}{1+\sigma}(h(x_1) - h_*) + \tfrac{1}{2}\langle 2\tfrac{A_1}{1+\sigma}g_1; x_* - x_0\rangle + \langle \tfrac{A_1}{1+\sigma}g_1; \lambda_1(g_1 + e_1)\rangle \\
& + \tfrac{A_1}{\sigma(1+\sigma)}[\tfrac{\lambda_1}{2}\|e_1\|^2 - \tfrac{\lambda_1\sigma^2}{2}\|e_1 + g_1\|^2] \\
= & \tfrac{A_1}{1+\sigma}(h(x_1) - h_*) + \tfrac{1}{4}\|x_* - x_0 + 2\tfrac{A_1}{1+\sigma}g_1\|^2 - \tfrac{1}{4}\|x_* - x_0\|^2 - \|\tfrac{A_1}{1+\sigma}g_1\|^2 \\
& + \langle \tfrac{A_1}{1+\sigma}g_1; \lambda_1(g_1 + e_1)\rangle + \tfrac{A_1}{\sigma(1+\sigma)}[\tfrac{\lambda_1}{2}\|e_1\|^2 - \tfrac{\lambda_1\sigma^2}{2}\|e_1 + g_1\|^2] \\
= & \tfrac{A_1}{1+\sigma}(h(x_1) - h_*) + \tfrac{1}{4}\|x_* - x_0 + 2\tfrac{A_1}{1+\sigma}g_1\|^2 - \tfrac{1}{4}\|x_* - x_0\|^2 + \tfrac{A_1\lambda_1(1-\sigma)}{2\sigma}\|e_1\|^2 \\
& + \tfrac{A_1\lambda_1(1-\sigma)}{1+\sigma}\langle g_1; e_1\rangle + \tfrac{A_1}{1+\sigma}\left(-\tfrac{A_1}{1+\sigma} + \lambda_1 - \tfrac{\lambda_1\sigma}{2}\right)\|g_1\|^2 \\
= & \tfrac{A_1}{1+\sigma}(h(x_1) - h_*) + \tfrac{1}{4}\|x_* - x_0 + 2\tfrac{A_1}{1+\sigma}g_1\|^2 - \tfrac{1}{4}\|x_* - x_0\|^2 \\
& + \tfrac{A_1\lambda_1(1-\sigma)}{2\sigma}\|e_1 + \tfrac{\sigma}{1+\sigma}g_1\|^2 + \tfrac{A_1}{1+\sigma}\left(-\tfrac{A_1}{1+\sigma} + \lambda_1 - \tfrac{\lambda_1\sigma}{2} - \tfrac{\lambda_1(1-\sigma)\sigma}{2(1+\sigma)}\right)\|g_1\|^2 \\
= & \tfrac{A_1}{1+\sigma}(h(x_1) - h_*) + \tfrac{1}{4}\|x_* - x_0 + 2\tfrac{A_1}{1+\sigma}g_1\|^2 - \tfrac{1}{4}\|x_* - x_0\|^2 \\
& + \tfrac{A_1\lambda_1(1-\sigma)}{2\sigma}\|e_1 + \tfrac{\sigma}{1+\sigma}g_1\|^2 + \tfrac{A_1}{1+\sigma}\left(\tfrac{\lambda_1 - A_1}{1+\sigma}\right)\|g_1\|^2 \\
= & \tfrac{A_1}{1+\sigma}(h(x_1) - h_*) + \tfrac{1}{4}\|x_* - x_0 + 2\tfrac{A_1}{1+\sigma}g_1\|^2 - \tfrac{1}{4}\|x_* - x_0\|^2 + \tfrac{A_1\lambda_1(1-\sigma)}{2\sigma}\|e_1 + \tfrac{\sigma}{1+\sigma}g_1\|^2
\end{aligned}
$$

where we used in the last line that $A_1 = \lambda_1$.

Now, assuming the weighted sum can be reformulated as the desired inequality for $N = k$, that is:

$$
\begin{aligned}
0 \geq & \tfrac{A_k}{1+\sigma}(h(x_k) - h_*) - \tfrac{1}{4}\|x_* - x_0\|^2 + \tfrac{1}{4}\|x_* - x_0 + \tfrac{2}{1+\sigma}\sum_{i=1}^{k}(A_i - A_{i-1})g_i\|^2 \\
& + \tfrac{(1-\sigma)}{2\sigma}\sum_{i=1}^{k} A_i\lambda_i\|e_i + \tfrac{\sigma}{1+\sigma}g_i\|^2,
\end{aligned}
$$

let us prove it also holds true for $N = k + 1$. Noticing that the weighted sum for $k + 1$ is exactly the weighted sum for $k$ (which can be reformulated as desired, through our induction hypothesis) with $4$ additional inequalities, we get the following valid inequality

$$
\begin{aligned}
0 \geq & \tfrac{A_k}{1+\sigma}(h(x_k) - h_*) - \tfrac{1}{4}\|x_* - x_0\|^2 + \tfrac{1}{4}\|x_* - x_0 + \tfrac{2}{1+\sigma}\sum_{i=1}^{k}(A_i - A_{i-1})g_i\|^2 \\
& + \tfrac{(1-\sigma)}{2\sigma}\sum_{i=1}^{k} A_i\lambda_i\|e_i + \tfrac{\sigma}{1+\sigma}g_i\|^2 \\
& + \tfrac{A_{k+1} - A_k}{1+\sigma}[h(u_{k+1}) - h_* + \langle g_{k+1}; x_* - u_{k+1}\rangle] \\
& + \tfrac{(1-\sigma)A_{k+1}}{(1+\sigma)\sigma}[h(u_{k+1}) - h(x_{k+1}) + \langle g_{k+1}; x_{k+1} - u_{k+1}\rangle] \\
& + \tfrac{A_k}{1+\sigma}[h(u_{k+1}) - h(x_k) + \langle g_{k+1}; x_k - u_{k+1}\rangle] \\
& + \tfrac{A_{k+1}}{(1+\sigma)\sigma}\big[\tfrac{\lambda_{k+1}}{2}\|e_{k+1}\|^2 - \tfrac{\lambda_{k+1}\sigma^2}{2}\|e_{k+1} + g_{k+1}\|^2 \\
& + h(x_{k+1}) - h(u_{k+1}) - \langle g_{k+1}; x_{k+1} - u_{k+1}\rangle\big].
\end{aligned}
$$

By regrouping all function values we get the following simplification:

$$\begin{aligned}
&[\tfrac{A_k}{1+\sigma} - \tfrac{A_k}{1+\sigma}]h(x_k) + \tfrac{A_{k+1}}{1+\sigma}[\tfrac{1}{\sigma} - \tfrac{1-\sigma}{\sigma}](h(x_{k+1}) - h_*) \\
&+ \tfrac{1}{1+\sigma}[A_{k+1} - A_k + \tfrac{1-\sigma}{\sigma}A_{k+1} + A_k - \tfrac{1}{\sigma}A_{k+1}]h(u_{k+1}) \\
=&\tfrac{A_{k+1}}{1+\sigma}(h(x_{k+1}) - h_*),
\end{aligned}$$

where $h(x_k)$ and $h(u_{k+1})$ disappear. The remaining inequality is therefore

$$\begin{aligned}
0 \geq&\tfrac{A_{k+1}}{1+\sigma}(h(x_{k+1}) - h_*) - \tfrac{1}{4}\|x_0 - x_*\|^2 + \tfrac{1}{4}\|x_* - x_0 + \tfrac{2}{1+\sigma}\sum_{i=1}^{k}(A_i - A_{i-1})g_i\|^2 \\
&+ \tfrac{(1-\sigma)}{2\sigma}\sum_{i=1}^{k} A_i\lambda_i\|e_i + \tfrac{\sigma}{1+\sigma}g_i\|^2 + \tfrac{A_{k+1}\lambda_{k+1}}{2(1+\sigma)\sigma}[\|e_{k+1}\|^2 - \sigma^2\|e_{k+1} + g_{k+1}\|^2] \\
&+ \tfrac{1}{1+\sigma}\langle g_{k+1}; (A_{k+1} - A_k)(x_* - u_{k+1}) - A_{k+1}(x_{k+1} - u_{k+1})\rangle + A_k(x_k - u_{k+1}) \\
=&\tfrac{A_{k+1}}{1+\sigma}(h(x_{k+1}) - h_*) - \tfrac{1}{4}\|x_0 - x_*\|^2 + \tfrac{1}{4}\|x_* - x_0 + \tfrac{2}{1+\sigma}\sum_{i=1}^{k}(A_i - A_{i-1})g_i\|^2 \\
&+ \tfrac{(1-\sigma)}{2\sigma}\sum_{i=1}^{k} A_i\lambda_i\|e_i + \tfrac{\sigma}{1+\sigma}g_i\|^2 + \tfrac{A_{k+1}\lambda_{k+1}}{2(1+\sigma)\sigma}[\|e_{k+1}\|^2 - \sigma^2\|e_{k+1} + g_{k+1}\|^2] \\
&+ \tfrac{1}{1+\sigma}\langle g_{k+1}; (A_{k+1} - A_k)x_* - A_{k+1}x_{k+1} + A_k x_k\rangle.
\end{aligned} \tag{5.30}$$

Then, by using (5.29), one can observe that

$$\begin{aligned}
A_{k+1}x_{k+1} =&\tfrac{A_{k+1}\lambda_{k+1}}{A_{k+1} - A_k}\left(x_0 - \tfrac{2}{1+\sigma}\sum_{i=1}^{k}(A_i - A_{i-1})g_i\right) + \left(A_{k+1} - \tfrac{A_{k+1}\lambda_{k+1}}{A_{k+1} - A_k}\right)x_k \\
&- A_{k+1}\lambda_{k+1}(g_{k+1} + e_{k+1}) \\
=&(A_{k+1} - A_k)\left(x_0 - \tfrac{2}{1+\sigma}\sum_{i=1}^{k}(A_i - A_{i-1})g_i\right) + A_k x_k - A_{k+1}\lambda_{k+1}(g_{k+1} + e_{k+1}),
\end{aligned}$$

and by re-injecting this inside the last line of (5.30), we get

$$\begin{aligned}
0 \geq&\tfrac{A_{k+1}}{1+\sigma}(h(x_{k+1}) - h_*) - \tfrac{1}{4}\|x_0 - x_*\|^2 + \tfrac{1}{4}\|x_* - x_0 + \tfrac{2}{1+\sigma}\sum_{i=1}^{k}(A_i - A_{i-1})g_i\|^2 \\
&+ \tfrac{(1-\sigma)}{2\sigma}\sum_{i=1}^{k} A_i\lambda_i\|e_i + \tfrac{\sigma}{1+\sigma}g_i\|^2 + \tfrac{A_{k+1}\lambda_{k+1}}{2(1+\sigma)\sigma}[\|e_{k+1}\|^2 - \sigma^2\|e_{k+1} + g_{k+1}\|^2] \\
&+ \tfrac{1}{1+\sigma}\langle (A_{k+1} - A_k)g_{k+1}; x_* - x_0 + \tfrac{2}{1+\sigma}\sum_{i=1}^{k}(A_i - A_{i-1})g_i\rangle \\
&+ \tfrac{A_{k+1}\lambda_{k+1}}{1+\sigma}\langle g_{k+1}; (g_{k+1} + e_{k+1})\rangle.
\end{aligned}$$

We can then proceed in a similar manner as in the case $k = 1$ for factorizing the quadratic terms,

$$0 \geq \tfrac{A_{k+1}}{1+\sigma}(h(x_{k+1}) - h_*) - \tfrac{1}{4}\|x_0 - x_*\|^2 + \tfrac{1}{4}\|x_* - x_0 + \tfrac{2}{1+\sigma}\sum_{i=1}^{k+1}(A_i - A_{i-1})g_i\|^2$$

$$
\begin{aligned}
&+ \tfrac{(1-\sigma)}{2\sigma}\sum_{i=1}^{k} A_i\lambda_i\|e_i + \tfrac{\sigma}{1+\sigma}g_i\|^2 + \tfrac{A_{k+1}\lambda_{k+1}}{2(1+\sigma)\sigma}[\|e_{k+1}\|^2 - \sigma^2\|e_{k+1}+g_{k+1}\|^2] \\
&- \tfrac{(A_{k+1}-A_k)^2}{(1+\sigma)^2}\|g_{k+1}\|^2 + \tfrac{A_{k+1}\lambda_{k+1}}{1+\sigma}\langle g_{k+1}; (g_{k+1}+e_{k+1})\rangle \\
=&\tfrac{A_{k+1}}{1+\sigma}(h(x_{k+1}) - h_*) - \tfrac{1}{4}\|x_0 - x_*\|^2 + \tfrac{1}{4}\|x_* - x_0 + \tfrac{2}{1+\sigma}\sum_{i=1}^{k+1}(A_i - A_{i-1})g_i\|^2 \\
&+ \tfrac{(1-\sigma)}{2\sigma}\sum_{i=1}^{k} A_i\lambda_i\|e_i + \tfrac{\sigma}{1+\sigma}g_i\|^2 + \tfrac{A_{k+1}\lambda_{k+1}(1-\sigma)}{2\sigma}\|e_{k+1} + \tfrac{\sigma}{1+\sigma}g_{k+1}\|^2 \\
&+ [\tfrac{A_{k+1}\lambda_{k+1}}{(1+\sigma)} - \tfrac{A_{k+1}\lambda_{k+1}\sigma}{2(1+\sigma)} - \tfrac{(A_{k+1}-A_k)^2}{(1+\sigma)^2} - \tfrac{A_{k+1}\lambda_{k+1}\sigma(1-\sigma)}{2(1+\sigma)^2}]\|g_{k+1}\|^2 \\
=&\tfrac{A_{k+1}}{1+\sigma}(h(x_{k+1}) - h_*) - \tfrac{1}{4}\|x_0 - x_*\|^2 + \tfrac{1}{4}\|x_* - x_0 + \tfrac{2}{1+\sigma}\sum_{i=1}^{k+1}(A_i - A_{i-1})g_i\|^2 \\
&+ \tfrac{(1-\sigma)}{2\sigma}\sum_{i=1}^{k+1} A_i\lambda_i\|e_i + \tfrac{\sigma}{1+\sigma}g_i\|^2 + \tfrac{A_{k+1}\lambda_{k+1}}{(1+\sigma)}[1 - \tfrac{\sigma}{2} - \tfrac{1}{(1+\sigma)} - \tfrac{\sigma(1-\sigma)}{2(1+\sigma)}]\|g_{k+1}\|^2 \\
=&\tfrac{A_{k+1}}{1+\sigma}(h(x_{k+1}) - h_*) - \tfrac{1}{4}\|x_0 - x_*\|^2 + \tfrac{1}{4}\|x_* - x_0 + \tfrac{2}{1+\sigma}\sum_{i=1}^{k+1}(A_i - A_{i-1})g_i\|^2 \\
&+ \tfrac{(1-\sigma)}{2\sigma}\sum_{i=1}^{k+1} A_i\lambda_i\|e_i + \tfrac{\sigma}{1+\sigma}g_i\|^2,
\end{aligned}
$$

since $1 - \frac{\sigma}{2} - \frac{1}{1+\sigma} - \frac{\sigma(1-\sigma)}{2(1+\sigma)} = 0$ and this concludes the proof. ■

# Chapter 6

# Some Inexact Proximal Algorithms and their Analyses

In this shorter chapter, we provide a simple version of an accelerated forward-backward method (a.k.a. Nesterov's accelerated proximal gradient method) possibly relying on approximate proximal operators and allowing to exploit strong convexity of the objective function. The method supports both relative and absolute errors, and its behavior is illustrated on a set of standard numerical experiments.

Using the same developments, we further provide a version of the *accelerated hybrid proximal extragradient* (A-HPE) method of Monteiro and Svaiter [2013] possibly exploiting strong convexity of the objective function. Finally, we provide a worst-case analysis of a partially inexact Douglas-Rachford algorithm due to Eckstein and Yao [2018].

The inexact forward-backward algorithm and the strongly convex A-HPE method were obtained using the developments presented in Chapter 5, but are presented separately as we believe they are of independent interest.

**Contributions** We can summarize our contributions as follows.

(i) We provide and analyze an inexact accelerated forward-backward method possibly supporting strongly convex objectives and possibly mixing relative and absolute error terms.

(ii) We provide and analyze an extension of the *accelerated hybrid proximal extragradient* method from Monteiro and Svaiter [2013] possibly supporting strongly convex objectives.

(iii) We present worst-case convergence bounds for the partially inexact Douglas-Rachford algorithm from Eckstein and Yao [2018] in a strongly convex setting.

**Organization** This note is organized as follows. First, we provide a short introduction on algorithm relying on inexact proximal computations in Section 6.1. Then, we give some base results and notations in Section 6.2. We provide the inexact accelerated forward-backward in Section 6.3, along with a worst-case analysis relying on a standard Lyapunov argument (for which we provide symbolic notebooks, helping the reader reproducing the algebraic part of the proof without pain). Numerical experiments illustrating the practical behavior of the method are then provided in Section 6.4. After that, Section 6.5 shows how to slightly improve the proof for obtaining an accelerated hybrid proximal

extragradient method [Monteiro and Svaiter, 2013], specifically dealing with nonsmooth strongly convex minimization. Finally, we analyze the partially inexact Douglas-Rachford splitting algorithm from Eckstein and Yao [2018] in Section 6.6. We draw some conclusions in Section 6.7.

## 6.1 Introduction

In this work, we consider a standard composite convex minimization problem of the form

$$\min_{x\in\mathbb{R}^d} \{F(x) \equiv f(x) + g(x)\}, \tag{6.1}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is a $L$-smooth convex function (with $0 < L < \infty$), and $g : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is a proper closed convex function. In addition, we allow either $f$ or $g$ to be possibly $\mu$-strongly convex. In this setting, we propose an inexact accelerated forward-backward method for solving (6.1) relying on the access to the gradient of $f$, and to an iterative routine for approximating the proximal operator of $g$.

**Relation to previous works** When the proximal operator of $g$ is readily available, the method presented below becomes a variant of standard accelerated (or fast) forward-backward (or proximal gradient) methods for convex minimization, see e.g., Nesterov [2013], Beck and Teboulle [2009], and the introductory survey by Aspremont et al. [2021].

Purely backward versions ($f = 0$) emerged earlier from the works of Güler [1992] and Monteiro and Svaiter [2013], Salzo and Villa [2012], whereas the first purely forward version ($g = 0$) was developed by Nesterov [1983]. The first inexact versions of accelerated forward-backward methods that we are aware of were presented in Schmidt et al. [2011], Villa et al. [2013], Jiang et al. [2012], whereas versions with relative errors appeared more recently in Millán and Machado [2019], Bello-Cruz et al. [2020]. In contrast, our method allows handling different types of error (namely absolute and relative errors of different types), while allowing to exploit strong convexity of $f$ or $g$—see e.g. Nesterov [2004, 2013], Chambolle and Pock [2016], for original analyses in the strongly convex case, when the proximal operator of $g$ is readily available. The same developments allow obtaining a strongly convex version of the accelerated hybrid proximal extragradient method (A-HPE), in the spirit of Monteiro and Svaiter [2013].

The notion of an "approximate proximal point" used in this note (see Section 6.2.2) was used in a few previous works, starting with the hybrid extragradient method [Solodov and Svaiter, 1999a, 2000a]. It was also used for its accelerated version [Monteiro and Svaiter, 2013] and in the context of another forward-backward splitting method [Millán and Machado, 2019]. In these works, the primal-dual requirement is presented under a different formulation involving the notion of $\varepsilon$-subdifferentials [Brøndsted and Rockafellar, 1965, Section 3] (or $\varepsilon$-enlargement in the context of monotone operators [Burachik et al., 1997, 1998]). We refer the reader to Section 5.2 of Chapter 5 for a survey on common notions of "approximate proximal point" used in the literature.

**Notations** We refer to classical textbooks [Rockafellar, 1996, Hiriart-Urruty and Lemaréchal, 2013] for standard elements of convex analysis. We use the notation $\mathcal{F}_{0,\infty}(\mathbb{R}^d)$ to denote the set of closed convex proper function on $\mathbb{R}^d$. The corresponding subset of closed convex proper functions that are $\mu$-strongly convex and $L$-smooth (with $0 \le \mu < L \le \infty$) is denoted $\mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$. That is, $h \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$ if and only if

- ($\mu$-strong convexity) $\forall x, y \in \mathbb{R}^d$, $s_h(x) \in \partial h(x)$, $s_h(y) \in \partial h(y)$, it holds $\|s_h(x) - s_h(y)\| \ge \mu\|x - y\|$,

- ($L$-smoothness) $\forall x, y \in \mathbb{R}^d$, $s_h(x) \in \partial h(x)$, $s_h(y) \in \partial h(y)$, it holds $\|s_h(x) - s_h(y)\| \leq L\|x - y\|$,

where $\partial h(x)$ denotes the subdifferential of $h$ at $x \in \mathbb{R}^d$. When $h \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$ with $L < \infty$, we use $\nabla h(x)$ to denote the unique element $\nabla h(x) \in \partial h(x)$ (i.e. the gradient of $h$ at $x$).

**Codes** For helping the reader reproducing the analytical results (via Mathematica notebooks) as well as numerical experiments, our code is available at

https://github.com/mathbarre/InexactStronglyConvexForwardBackward/tree/extended.

## 6.2 Background results

### 6.2.1 Smooth strongly convex functions

We recall some standard inequalities satisfied by smooth convex and strongly convex functions (see Chapter 3), which we use in the sequel for exploiting strong convexity and smoothness, see e.g. Nesterov [2004].

**Proposition 6.2.1** ($\mu$-strong convexity)**.** *Let $g \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$. For all $x, y \in \mathbb{R}^d$ and all $s_g(x) \in \partial g(x)$ it holds that*

$$g(y) \geq g(x) + \langle s_g(x), y - x\rangle + \tfrac{\mu}{2}\|x - y\|^2.$$

**Proposition 6.2.2** ($L$-smoothness & convexity)**.** *Let $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$ with $L < +\infty$. For all $x, y \in \mathbb{R}^d$ it holds that*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle + \tfrac{1}{2L}\|\nabla f(x) - \nabla f(y)\|^2.$$

In the sequel, the use of the inequalities provided by Proposition 6.2.1 and Proposition 6.2.2 are motivated by their *interpolation* (or extension) properties; that is, the analyses provided below were obtained following a principled approach to worst-case analyses of first-order methods, see e.g., Taylor et al. [2017c] or Chapter 5 specifically for the cases of methods relying on approximate proximal operations.

### 6.2.2 Proximal operations

The proximal operation is a base primitive that is widely used in modern optimization methods; it is a central building blocks in many optimization algorithms, see e.g., Parikh and Boyd [2014], Ryu and Boyd [2016]. The proximal operator of a function $g \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ with step size $\lambda \geq 0$ is defined as

$$\mathrm{prox}_{\lambda g}(z) = \operatorname*{argmin}_{x \in \mathbb{R}^d} \left\{ \lambda g(x) + \tfrac{1}{2}\|x - z\|^2 \right\}, \tag{6.2}$$

with $z \in \mathbb{R}^d$. When $g \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$, the proximal operation is well defined, and its solution is unique. A comprehensive list of cases where (6.2) has an analytical solution is provided in Chierchia et al. [2020]. In other cases, the proximal operator has to be approximated. For doing that, one can define the following primal and dual problems associated to the proximal operation

$$\min_{x \in \mathbb{R}^d} \{\Phi_{\mathrm{p}}(z;\, x) \equiv \lambda g(x) + \tfrac{1}{2}\|x - z\|^2\}, \tag{P}$$

$$\max_{v\in\mathbb{R}^d} \{\Phi_{\mathrm{d}}(z;\, v) \equiv -\lambda g^*(v) - \tfrac{1}{2}\|z-\lambda v\|^2 + \tfrac{1}{2}\|z\|^2\}, \tag{D}$$

where $g^* \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ is the Fenchel conjugate of $g$. Let us further note that $\mathrm{prox}_{\lambda g}(z)$ is the unique solution to (P), and that $\mathrm{prox}_{g^*/\lambda}(z/\lambda)$ is the unique solution of (D). In this context, the primal and dual solutions are linked by the well-known Moreau's identity $\mathrm{prox}_{\lambda g}(z) + \lambda\,\mathrm{prox}_{g^*/\lambda}(z/\lambda) = z$.

Under relatively weak conditions (such as $\mathrm{ri}(\mathbf{dom}\, g) \neq \emptyset$, see e.g., Chambolle and Pock [2016, Section 3.5]), strong duality holds between (P) and (D) and hence

$$\min_{x\in\mathbb{R}^d} \Phi_{\mathrm{p}}(z;\, x) = \max_{v\in\mathbb{R}^d} \Phi_{\mathrm{d}}(z;\, v).$$

Motivated by those elements, we use the quantity

$$\mathrm{PD}_{\lambda g}(x, v;\, z) = \Phi_{\mathrm{p}}(z;\, x) - \Phi_{\mathrm{d}}(z;\, v), \tag{PD}$$

for quantifying how well $(x, v)$ approximates the pair $(\mathrm{prox}_{\lambda g}(z), \mathrm{prox}_{g^*/\lambda}(z/\lambda))$, in the sequel.

### 6.2.3 A notion of approximate proximal point

In this section, we define the notion of approximate proximal point of $g \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ used throughout the chapter (see Section 6.1 §"Relation to previous works" for historical references for the case $\mu = 0$). This notion features two parameters: a tolerance and a lower bound on the strong convexity parameter of $g$ (possibly 0). The estimate of the strong convexity is used for relating proximal points of $g(\cdot)$ in terms of that of $g_\mu(\cdot) = g(\cdot) - \frac{\mu}{2}\|\cdot\|^2 \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$, and the tolerance is used for quantifying the quality of an approximate solution to the proximal problem on $g_\mu(\cdot)$, which simplifies the analyses below. More precisely, for $g \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$, it is relatively straightforward to verify that

$$\mathrm{prox}_{\lambda g}(z) = \mathrm{prox}_{\frac{\lambda}{1+\lambda\mu} g_\mu}\left(\tfrac{z}{1+\lambda\mu}\right),$$

with $g_\mu(x) = g(x) - \frac{\mu}{2}\|x\|^2$. This observation motivates the introduction of the following inexactness criterion.

**Definition 6.2.3.** *Let $\mu > 0$, $g \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$, and let $\lambda > 0$ be a step size and $\varepsilon \geq 0$ be a tolerance. For a triplet $(x, v, y) \in \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d$ we use the notation*

$$(x, v) \approx_{\varepsilon,\mu} \left(\mathrm{prox}_{\lambda g}(y), \mathrm{prox}_{\frac{g*}{\lambda}}\left(\tfrac{y}{\lambda}\right)\right),$$

*for denoting that*

$$\mathrm{PD}_{\frac{\lambda}{1+\mu\lambda} g_\mu}\left(x, v - \mu x;\, \tfrac{y}{1+\mu\lambda}\right) \leq \varepsilon,$$

*with $g_\mu(x) = g(x) - \frac{\mu}{2}\|x\|^2$ and* PD *the primal-dual gap of the proximal problem defined in* (PD).

In the following technical lemma, we provide an explicit expression for quantifying the quality of a triplet $(x, v, y) \in \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d$ in light of Definition 6.2.3.

**Lemma 6.2.4.** *Let $\mu \geq 0$, $g \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$, and let $\lambda > 0$ be a step size and $(x, v, z) \in \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d$. The following equality holds*

$$\begin{aligned}\mathrm{PD}_{\frac{\lambda}{1+\mu\lambda} g_\mu}(x, v - \mu x;\, \tfrac{z}{1+\mu\lambda}) = {} & \tfrac{1}{2(1+\lambda\mu)^2}\|x - z + \lambda v\|^2 \\ & + \tfrac{\lambda}{1+\lambda\mu}\left(g(x) - g(w) + \tfrac{\mu}{2}\|x - w\|^2 - \langle x - w, v\rangle\right),\end{aligned} \tag{6.3}$$

*with $g_\mu(\cdot) = g(\cdot) - \frac{\mu}{2}\|\cdot\|^2$ and $w \in \mathbb{R}^d$ satisfying $v - \mu x + \mu w \in \partial g(w)$ (i.e., $w \in \partial g^*_\mu(v - \mu x)$).*

*Proof.*

$$\begin{aligned}\mathrm{PD}_{\frac{\lambda}{1+\mu\lambda}g_\mu}(x, v-\mu x; \tfrac{z}{1+\mu\lambda}) &= \tfrac{1}{2}\|x - \tfrac{z}{1+\lambda\mu} + \tfrac{\lambda}{1+\lambda\mu}(v-\mu x)\|^2 \\ &\quad + \tfrac{\lambda}{1+\lambda\mu}\left(g_\mu(x) + g_\mu^*(v-\mu x) - \langle x, v-\mu x\rangle\right) \\ &= \tfrac{1}{2(1+\lambda\mu)^2}\|x - z + \lambda v\|^2 \\ &\quad + \tfrac{\lambda}{1+\lambda\mu}\left(g(x) + \tfrac{\mu}{2}\|x\|^2 + g_\mu^*(v-\mu x) - \langle x, v\rangle\right).\end{aligned} \tag{6.4}$$

In particular

$$g_\mu^*(v-\mu x) = \max_y \langle y, v-\mu x\rangle - g(y) + \tfrac{\mu}{2}\|y\|^2,$$

and by choosing $w \in \mathbb{R}^d$ such that $v - \mu x + \mu w \in \partial g(w)$ we get

$$g_\mu^*(v-\mu x) = \langle w, v-\mu x\rangle - g(w) + \tfrac{\mu}{2}\|w\|^2. \tag{6.5}$$

Finally, using the expression of $g_\mu^*(v-\mu x)$ in (6.4) leads to the desired results. ∎

In the next section, we present an inexact accelerated forward-backward method where inexactness in proximal computations are measured using the primal-dual criterion from Definition 6.2.3.

## 6.3 An inexact accelerated forward-backward method

In this section, we provide the main contribution of this work, namely Algorithm 6.3.1. This method aims at solving (6.1) when the gradient of $f$ is readily available and the proximal operator of $g$ can be efficiently approximated within a target precision (e.g., by an iterative method). It further allows to exploit $g$ to be $\mu$-strongly convex. In the case where $f$ is strongly convex, one can shift this strong convexity to $g$ instead (by removing the corresponding quadratic of $f$ and adding it to $g$). Of course, any under-approximation of $\mu$ can be used within the method.

The worst-case analysis is based on a simple Lyapunov (or potential) argument, following the now standard template for accelerated schemes as in Nesterov [1983], for which surveys are provided in e.g., Bansal and Gupta [2019], Wilson et al. [2021], and Aspremont et al. [2021, Chapter 4]. As a byproduct of the analysis, the method does not require an accurate estimate of the smoothness constant $L$, whose estimation is improved on the fly using standard backtracking tricks, similar in spirit with Nesterov [1983], Beck and Teboulle [2009].

The algorithm below builds on approximations of the forward-backward operator (with step sizes $\lambda_k$) of problem (6.1). More precisely, it relies on primal-dual pairs $(x_{k+1},\ v_{k+1})$ approximating the forward-backward operator evaluated at some iterates $y_k$, and satisfying

$$(x_{k+1},\ v_{k+1}) \approx_{\varepsilon_k,\mu} \left(\mathrm{prox}_{\lambda_k g}(y_k - \lambda_k \nabla f(y_k)), \mathrm{prox}_{\frac{g^*}{\lambda_k}}(\tfrac{y_k - \lambda_k \nabla f(y_k)}{\lambda_k})\right),$$

where $\varepsilon_k$ encodes some approximation level. In this work, this error term is parameterized by three sequences of nonnegative scalars $\{\sigma_k\}_k$, $\{\zeta_k\}_k$, $\{\xi_k\}_k$ that can be chosen by the user for possibly mixing both *relative* (or multiplicative) and *absolute* (or additive) error terms

$$\varepsilon_k = \tfrac{\sigma_k^2}{2(1+\lambda_k\mu)^2}\|x_{k+1} - y_k\|^2 + \tfrac{\zeta_k^2\lambda_k^2}{2(1+\lambda_k\mu)^2}\|v_{k+1} + \nabla f(y_k)\|^2 + \tfrac{\lambda_k\xi_k}{2(1+\lambda_k\mu)^2},$$

where $\{\xi_k\}_k$ parametrizes the absolute error term, and where $\{\sigma_k\}_k$ and $\{\zeta_k\}_k$ parametrize two types of relative errors. Of course, convergence properties of the algorithm depend on the choice of those

sequences of parameters, as provided in Corollary 6.3.4 and Corollary 6.3.5 below. Examples of simple rules for $\{\sigma_k\}_k$, $\{\zeta_k\}_k$, $\{\xi_k\}_k$ are provided in Section 6.4 (typically, $\{\sigma_k\}_k$, $\{\zeta_k\}_k$ can be chosen constant, whereas $\{\xi_k\}_k$ should be either identically 0 or decreasing fast enough).

Before going into the algorithm itself, let us mention that the backtracking line-search strategy (Btr) for estimating the smoothness constant builds on the condition

$$f(y_k) \geq f(x_{k+1}) + \langle \nabla f(x_{k+1}), y_k - x_{k+1} \rangle + \frac{\lambda_k}{2(1-\sigma_k^2)} \|\nabla f(y_k) - \nabla f(x_{k+1})\|^2, \qquad \text{(Smooth)}$$

where $x_k$'s and $y_k$'s are some iterates. In particular, picking $\lambda_k \in (0, \frac{1-\sigma_k^2}{L}]$ (hence depending on the true smoothness constant $L$) guarantees (Smooth) to be satisfied without backtracking, as, when $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$, Proposition 6.2.2 holds.

### 6.3.1 Algorithm

**An inexact accelerated forward-backward method** (Algorithm 6.3.1)

**Input:**

- Objective function: $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$, $g \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$, and $\mu \geq 0$.
- Initial point: $x_0 \in \mathbb{R}^d$.
- Initial step size: $\lambda_0 > 0$.
- Tolerance parameters: $\{\sigma_k\}_k$, $\{\zeta_k\}_k$ with $\sigma_k$, $\zeta_k \in [0, 1)$, and $\{\xi_k\}_k$ with $\xi_k \geq 0$.
- Backtracking parameters $0 < \alpha < 1$ and $\beta \geq 1$.

**Initialization:**

$z_0 = x_0$, $A_0 = 0$,

**Run:**

For $k = 0, 1, \ldots$:

$$\eta_k = (1 - \zeta_k^2)\lambda_k \qquad (6.6)$$

$$A_{k+1} = A_k + \frac{\eta_k + 2A_k\mu\eta_k + \sqrt{\eta_k^2 + 4\eta_k A_k(1+\eta_k\mu)(1+A_k\mu)}}{2}$$

$$y_k = x_k + \frac{(A_{k+1} - A_k)(A_k\mu + 1)}{A_{k+1} + A_k(2A_{k+1} - A_k)\mu}(z_k - x_k)$$

$$\varepsilon_k = \frac{\sigma_k^2}{2(1+\lambda_k\mu)^2}\|x_{k+1} - y_k\|^2 + \frac{\zeta_k^2\lambda_k^2}{2(1+\lambda_k\mu)^2}\|v_{k+1} + \nabla f(y_k)\|^2 + \frac{\lambda_k\xi_k}{2(1+\lambda_k\mu)^2}$$

$$(x_{k+1},\ v_{k+1}) \approx_{\varepsilon_k,\mu} \left(\mathrm{prox}_{\lambda_k g}(y_k - \lambda_k\nabla f(y_k)), \mathrm{prox}_{\frac{g^*}{\lambda_k}}\left(\frac{y_k - \lambda_k\nabla f(y_k)}{\lambda_k}\right)\right)$$

[If (Smooth) is not satisfied, set $\lambda_k \leftarrow \alpha\lambda_k$ and go back to step (6.6)] (Btr)

$$z_{k+1} = z_k + \frac{A_{k+1} - A_k}{1 + \mu A_{k+1}}\left(\mu(x_{k+1} - z_k) - (v_{k+1} + \nabla f(y_k))\right)$$

$$\lambda_{k+1} = \beta\lambda_k$$

End For

**Output:** $x_{k+1}$

**Remark 6.3.1** (Related methods)**.** *When the objective function is not strongly convex (i.e. $\mu = 0$), the update rules of Algorithm 6.3.1 are very similar to those of the accelerated inexact forward-backward methods from Millán and Machado [2019, Algorithm 3] (when $\zeta_k = 0$ and $\xi_k = 0$) or Bello-Cruz et al. [2020, Algorithm 2] (when $\sigma_k = 0$ and $\xi_k = 0$). Compared to those works in this setup, Algorithm 6.3.1 allows using both relative and absolute errors while having a backtracking strategy. Note also the similarities with some inexact FISTA [Villa et al., 2013, Schmidt et al., 2011], although these methods do not re-use explicitly the dual direction $v_{k+1}$ and focus only on absolute error terms (i.e., $\sigma_k = \zeta_k = 0$). Finally, when the computation of the proximal operator is exact, we recover one of the many variants of an accelerated forward-backward method; see for example Nesterov [2013], Tseng [2008], Beck and Teboulle [2009]; we refer to Aspremont et al. [2021, Chapter 4] and the references therein for further discussions on this topic.*

The following theorem contains the main (Lyapunov-based) ingredient of the worst-case analysis.

**Theorem 6.3.2.** *Let $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$, $g \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$, $F \equiv f + g$, $k \geq 0$, parameters $\sigma_k, \zeta_k \in [0, 1)$, $\xi_k \geq 0$ and some $\lambda_k > 0$ such that (Smooth) is satisfied. For any $x_k, z_k \in \mathbb{R}^d$, and $A_k \geq 0$, it holds that*

$$\begin{aligned} A_{k+1}(F(x_{k+1}) - F(x_*)) + \tfrac{1+\mu A_{k+1}}{2}\|z_{k+1} - x_*\|^2 \\ \leq A_k(F(x_k) - F(x_*)) + \tfrac{1+\mu A_k}{2}\|z_k - x_*\|^2 + \tfrac{A_{k+1}}{2}\xi_k, \end{aligned} \tag{6.7}$$

*with $x_* \in \operatorname{argmin}_x F(x)$, and where $z_{k+1}$, $x_{k+1}$ are constructed by one iteration of Algorithm 6.3.1.*

The proof of this Theorem is deferred to Section 6.3.2. The following (classical) corollary establishes that the growth rate of the sequence $\{A_k\}_k$ drives the convergence rate of the worst-case guarantee. Those factors $A_{k+1}$, controlling the convergence rate, were greedily chosen (as large as possible) while enforcing (6.7) to hold.

**Corollary 6.3.3.** *Let $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$, $g \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$ and $F \equiv f + g$. Let $x_0 \in \mathbb{R}^d$, $\lambda_0$ be a positive initial step size, $\alpha \in (0, 1)$ and $\beta \geq 1$ be some backtracking parameters, sequences (relative error parameters) $\{\sigma_k\}_k$, $\{\zeta_k\}_k$, satisfying $\sigma_k, \zeta_k \in [0, 1)$ and a sequence (absolute error parameters) $\{\xi_k\}_k$ with $\xi_k \geq 0$. Let $x_N \in \mathbb{R}^d$ be the output after $N \in \mathbb{N}^*$ iterations of Algorithm 6.3.1 on $F$ initiated at $x_0 \in \mathbb{R}^d$, it holds that*

$$F(x_N) - F(x_*) \leq \tfrac{1}{2A_N}\|x_0 - x_*\|^2 + \sum_{i=0}^{N-1} \tfrac{A_{i+1}}{2A_N}\xi_i,$$

*where $x_* \in \operatorname{argmin}_x F(x)$.*

*Proof.* We denote by $\Phi_k$ the quantity (a.k.a., the Lyapunov/potential function)

$$\Phi_k = A_k(F(x_k) - F(x_*)) + \tfrac{1+\mu A_k}{2}\|z_k - x_*\|^2,$$

for $k \geq 0$. Theorem 6.3.2 allows nesting the $\Phi_k$'s together as

$$\Phi_N \leq \Phi_{N-1} + \tfrac{A_N}{2}\xi_{N-1} \leq \ldots \leq \Phi_1 + \sum_{i=1}^{N-1} \tfrac{A_{i+1}}{2}\xi_i \leq \Phi_0 + \sum_{i=0}^{N-1} \tfrac{A_{i+1}}{2}\xi_i.$$

We reach the target conclusion using $A_N(F(x_N) - F(x_*)) \leq \Phi_N$, together with $z_0 = x_0$ and $A_0 = 0$. ∎

Let us note that when $\mu = 0$, we recover a composite version of the A-HPE method [Monteiro and Svaiter, 2013]. In that case, we can bound $A_k \geq \frac{1}{4}\left(\sum_{i=0}^{k-1}\sqrt{\eta_k}\right)^2 \geq \frac{\eta_{\min}}{4}k^2$, assuming the existence of some $\eta_{\min} \leq \eta_k$ for all $k \geq 0$. Such a lower bound on $\eta_k$ exists as soon as the parameters $\{\sigma_k\}_k$, $\{\zeta_k\}_k$ are well chosen (see for example Corollary 6.3.4 and Corollary 6.3.5 below), and due to the $L$-smoothness of the function. Similarly, when $\mu > 0$, $A_k$'s are growing exponentially as

$$\begin{aligned} A_{k+1} &= A_k + \frac{\eta_k + 2A_k\eta_k\mu + \sqrt{4\eta_k A_k(A_k\mu+1)(\eta_k\mu+1)+\eta_k^2}}{2} \\ &\geq A_k(1+\eta_k\mu) + A_k\sqrt{\eta_k\mu(1+\eta_k\mu)} \\ &= A_k/\left(1-\sqrt{\tfrac{\eta_k\mu}{1+\eta_k\mu}}\right), \end{aligned}$$

with $A_1 > 0$, reaching $1/A_k \leq \eta_{\min}\left(1-\sqrt{\frac{\eta_{\min}\mu}{1+\eta_{\min}\mu}}\right)^{k-1}$ assuming again the existence of some $\eta_{\min} \leq \eta_k$ for all $k \geq 0$. The following corollaries provide more precise convergence bounds for Algorithm 6.3.1, by quantifying the growth rate of the $A_k$'s, for some particular choices of parameters $\{\sigma_k\}_k$ (constant), $\{\zeta_k\}_k$ (constant), and $\{\xi_k\}_k$ (parameterized function of $k$), linking the behavior of the decrease rate of the absolute errors $\xi_k$ with the convergence bound.

**Corollary 6.3.4.** *Let $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$, $g \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$ and $F \equiv f + g$. Let $x_0 \in \mathbb{R}^d$, $\lambda_0$ be an initial positive step size, $\alpha \in (0,1)$ and $\beta \geq 1$ be some backtracking parameters, sequences (relative error parameters) $\sigma_k = \sigma$, $\zeta_k = \zeta$ with $\sigma, \zeta \in [0,1)$ and a sequence (absolute error parameters) $\xi_k = C\rho^k$ with $C, \rho > 0$. Let $x_N \in \mathbb{R}^d$ be the output after $N \in \mathbb{N}^*$ iterations of Algorithm 6.3.1 on $F$ initiated at $x_0 \in \mathbb{R}^d$, it holds that*

$$F(x_N) - F(x_*) \leq \frac{1}{2\eta}\left(1-\sqrt{\tfrac{\eta\mu}{1+\eta\mu}}\right)^{N-1}\|x_0 - x_*\|^2 + \begin{cases} \frac{C}{2(1-\sqrt{\frac{\eta\mu}{1+\eta\mu}}-\rho)}\left(1-\sqrt{\frac{\eta\mu}{1+\eta\mu}}\right)^N & \text{if } \rho < \tau, \\ \frac{1}{2}CN\left(1-\sqrt{\frac{\eta\mu}{1+\eta\mu}}\right)^{N-1} & \text{if } \rho = \tau, \\ \frac{C}{2(\rho-1+\sqrt{\frac{\eta\mu}{1+\eta\mu}})}\rho^N & \text{if } \rho > \tau, \end{cases}$$

*for some $\eta = \min_{i=0,\dots,N-1}\eta_i \geq \eta_{\min} = (1-\zeta^2)\min\left(\lambda_0, \frac{\alpha(1-\sigma^2)}{L}\right)$ and where $\tau = 1-\sqrt{\frac{\eta\mu}{1+\eta\mu}}$ and $x_* \in \operatorname{argmin}_x F(x)$.*

*Proof.* Starting from the conclusion of Corollary 6.3.3, we obtain the desired result using classical properties of geometric sums along with $A_k \leq \left(1-\sqrt{\frac{\eta\mu}{1+\eta\mu}}\right)^{N-k}A_N$ where $\eta = \min_{i=0,\dots,N-1}\eta_i \geq \eta_{\min}$. ■

When $\mu = 0$, the proof is still valid, and $\frac{1}{A_N} = O(N^{-2})$. In particular, we recover the same rates as those of Villa et al. [2013, Theorem 4.4] (who used the particular choice $v_{k+1} = \frac{y_k - \lambda_k\nabla f(y_k) - x_{k+1}}{\lambda_k}$).

**Corollary 6.3.5.** *Let $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$, $g \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$ and $F \equiv f + g$. Let $x_0 \in \mathbb{R}^d$, $\lambda_0$ be an initial positive step size and sequences (relative error parameters) $\sigma_k = \sigma$, $\zeta_k = \zeta$ with $\sigma, \zeta \in [0,1)$. Let $x_N \in \mathbb{R}^d$ denote the output after $N \in \mathbb{N}^*$ iterations of Algorithm 6.3.1 on $F$ initiated at $x_0 \in \mathbb{R}^d$.*

- *We further let $\alpha \in (0,1)$ and $\beta = 1$ be the backtracking parameters, and a sequence (absolute error parameters) $\xi_k = C(k+1)^{-q}$ with $C, q \geq 0$. It holds that*

$$F(x_N) - F(x_*) \leq \frac{2}{\eta_{\min}N^2}\|x_0 - x_*\|^2 + \begin{cases} 2C\frac{\eta_{\max}}{\eta_{\min}}\frac{\left(\sum_{k=0}^{\infty}(k+1)^{2-q}\right)}{N^2} & \text{if } q > 3, \\ 2C\frac{\eta_{\max}}{\eta_{\min}}\frac{(1+\ln(N))}{N^2} & \text{if } q = 3, \\ 2C\frac{\eta_{\max}}{\eta_{\min}}\left(\frac{1}{N^2} + \frac{1}{(3-q)N^{q-1}}\right) & \text{if } 1 < q < 3, \end{cases}$$

*with* $\eta_{\min} = (1-\zeta^2)\min(\lambda_0, \frac{\alpha(1-\sigma^2)}{L})$, $\eta_{\max} = (1-\zeta^2)\max(\lambda_0, \frac{(1-\sigma^2)}{L})$ *and where* $x_* \in \operatorname{argmin}_x F(x)$.

- *We further let* $\alpha \in (0,1)$ *and* $\beta \geq 1$ *be the backtracking parameters, and a sequence* $\xi_k = 0$ *(no absolute error). It holds that*

$$F(x_N) - F(x_*) \leq \frac{2}{\eta_{\min} N^2}\|x_0 - x_*\|^2,$$

*where* $\eta_{\min} = (1-\zeta^2)\min(\lambda_0, \frac{\alpha(1-\sigma^2)}{L})$ *and* $x_* \in \operatorname{argmin}_x F(x)$.

*Proof.* Starting from the conclusion of Corollary 6.3.3, we obtain the desired result in the case $\beta = 1$ using comparisons of sums with integrals along with the bounds $\frac{\eta_{\min}}{4}k^2 \leq A_k \leq \eta_{\max}k^2$. In the second case, where $\beta \geq 1$ and $\xi_k = 0$, the target result follows from $\frac{\eta_{\min}}{4}k^2 \leq A_k$. ■

### 6.3.2 Proof of Theorem 6.3.2

The following proof is presented in a purely algebraic form consisting in a weighted sum of inequalities satisfied by the functions $f$ and $g$ as well as inexactness requirements. Indeed, it has been obtained from a dual certificate of a performance estimation problem (see Section 5.3 of Chapter 5 for more details on performance estimation in the context of inexact proximal operations). As mentioned in Section 6.1, the algebraic equivalences stated below can be verified either by hand or with help of Mathematica notebooks (see Section 6.1, §"Codes").

*Proof.* Let $w_{k+1} \in \mathbb{R}^d$ such that $v_{k+1} - \mu x_{k+1} + \mu w_{k+1} \in \partial g(w_{k+1})$. Using (6.3), this leads to

$$\begin{aligned}\operatorname{PD}_{\frac{\lambda_k}{1+\mu\lambda_k} g_\mu}(x_{k+1}, v_{k+1} - \mu x_{k+1}; \tfrac{y_k - \lambda_k \nabla f(y_k)}{1+\mu\lambda_k}) = \tfrac{1}{2(1+\lambda_k\mu)^2}\|x_{k+1} - y_k + \lambda_k(v_{k+1} + \nabla f(y_k))\|^2 \\ + \tfrac{\lambda_k}{1+\lambda_k\mu}\Big(g(x_{k+1}) - g(w_{k+1}) \\ + \tfrac{\mu}{2}\|x_{k+1} - w_{k+1}\|^2 - \langle x_{k+1} - w_{k+1}, v_{k+1}\rangle\Big),\end{aligned}$$

with $g_\mu(\cdot) = g(\cdot) - \frac{\mu}{2}\|\cdot\|^2$.

The proof consists in performing a weighted sum of the following inequalities:

- strong convexity of $g$ between $w_{k+1}$ and $x_*$ with weight $\nu_1 = A_{k+1} - A_k$

$$g(x_*) \geq g(w_{k+1}) + \langle v_{k+1} - \mu x_{k+1} + \mu w_{k+1}, x_* - w_{k+1}\rangle + \tfrac{\mu}{2}\|w_{k+1} - x_*\|^2,$$

- strong convexity of $g$ between $w_{k+1}$ and $x_k$ with weight $\nu_2 = A_k$

$$g(x_k) \geq g(w_{k+1}) + \langle v_{k+1} - \mu x_{k+1} + \mu w_{k+1}, x_k - w_{k+1}\rangle + \tfrac{\mu}{2}\|w_{k+1} - x_k\|^2,$$

- strong convexity of $g$ between $w_{k+1}$ and $x_{k+1}$ with weight $\nu_3 = A_{k+1}\lambda_k\mu$

$$g(x_{k+1}) \geq g(w_{k+1}) + \langle v_{k+1} - \mu x_{k+1} + \mu w_{k+1}, x_{k+1} - w_{k+1}\rangle + \tfrac{\mu}{2}\|w_{k+1} - x_{k+1}\|^2,$$

- convexity of $f$ between $y_k$ and $x_*$ with weight $\nu_4 = A_{k+1} - A_k$

$$f(x_*) \geq f(y_k) + \langle \nabla f(y_k), x_* - y_k\rangle,$$

- convexity of $f$ between $y_k$ and $x_k$ with weight $\nu_5 = A_k$

$$f(x_k) \geq f(y_k) + \langle \nabla f(y_k), x_k - y_k \rangle,$$

- convexity and $\frac{1-\sigma_k^2}{\lambda_k}$-smoothness of $f$ between $x_{k+1}$ and $y_k$ required by (Smooth) with weight $\nu_6 = A_{k+1}$

$$f(y_k) \geq f(x_{k+1}) + \langle \nabla f(x_{k+1}), y_k - x_{k+1} \rangle + \tfrac{\lambda_k}{2(1-\sigma_k^2)} \|\nabla f(y_k) - \nabla f(x_{k+1})\|^2,$$

- approximation requirement on $x_{k+1}$ with weight $\nu_7 = \frac{A_{k+1}}{\lambda_k}$

$$\begin{aligned}\tfrac{\sigma_k^2}{2}\|x_{k+1} - y_k\|^2 + \tfrac{\zeta_k^2\lambda_k^2}{2}\|v_{k+1} + \nabla f(y_k)\|^2 + \tfrac{\lambda_k}{2}\xi_k \geq \tfrac{1}{2}\|x_{k+1} - y_k + \lambda_k(v_{k+1} + \nabla f(y_k))\|^2 \\ + \lambda_k(1+\lambda_k\mu)\Big(g(x_{k+1}) - g(w_{k+1}) \\ + \tfrac{\mu}{2}\|x_{k+1} - w_{k+1}\|^2 \\ - \langle x_{k+1} - w_{k+1}, v_{k+1} \rangle\Big).\end{aligned}$$

The weighted sum can be written as

$$\begin{aligned}0 \geq\ & \nu_1\left[g(w_{k+1}) - g(x_*) + \langle v_{k+1} - \mu x_{k+1} + \mu w_{k+1}, x_* - w_{k+1}\rangle + \tfrac{\mu}{2}\|w_{k+1} - x_*\|^2\right] \\ & + \nu_2\left[g(w_{k+1}) - g(x_k) + \langle v_{k+1} - \mu x_{k+1} + \mu w_{k+1}, x_k - w_{k+1}\rangle + \tfrac{\mu}{2}\|w_{k+1} - x_k\|^2\right] \\ & + \nu_3\Big[g(w_{k+1}) - g(x_{k+1}) + \langle v_{k+1} - \mu x_{k+1} + \mu w_{k+1}, x_{k+1} - w_{k+1}\rangle \\ & \quad + \tfrac{\mu}{2}\|w_{k+1} - x_{k+1}\|^2\Big] \\ & + \nu_4\left[f(y_k) - f(x_*) + \langle \nabla f(y_k), x_* - y_k\rangle\right] + \nu_5\left[f(y_k) - f(x_k) + \langle \nabla f(y_k), x_k - y_k\rangle\right] \\ & + \nu_6\left[f(x_{k+1}) - f(y_k) + \langle \nabla f(x_{k+1}), y_k - x_{k+1}\rangle + \tfrac{\lambda_k}{2(1-\sigma_k^2)}\|\nabla f(y_k) - \nabla f(x_{k+1})\|^2\right] \\ & + \nu_7\Big[\lambda_k(1+\lambda_k\mu)\Big(g(x_{k+1}) - g(w_{k+1}) + \tfrac{\mu}{2}\|x_{k+1} - w_{k+1}\|^2 - \langle x_{k+1} - w_{k+1}, v_{k+1}\rangle\Big) \\ & \quad + \tfrac{1}{2}\|x_{k+1} - y_k + \lambda_k(v_{k+1} + \nabla f(y_k))\|^2 - \tfrac{\sigma_k^2}{2}\|x_{k+1} - y_k\|^2 - \tfrac{\lambda_k}{2}\xi_k \\ & \quad - \tfrac{\zeta_k^2\lambda_k^2}{2}\|v_{k+1} + \nabla f(y_k)\|^2\Big].\end{aligned} \tag{6.8}$$

Substituting $y_k$ and $z_{k+1}$ in the weighted sum, that is

$$\begin{aligned} y_k &= x_k + \tfrac{(A_{k+1} - A_k)(1+\mu A_k)}{A_{k+1} + \mu A_k(2A_{k+1} - A_k)}(z_k - x_k) \\ z_{k+1} &= z_k + \tfrac{A_{k+1} - A_k}{1+\mu A_{k+1}}\left(\mu(x_{k+1} - z_k) - (v_{k+1} + \nabla f(y_k))\right), \end{aligned}$$

(6.8) is equivalently reformulated as

$$\begin{aligned}& A_{k+1}(F(x_{k+1}) - F(x_*)) + \tfrac{1+\mu A_{k+1}}{2}\|z_{k+1} - x_*\|^2 \\ & \leq A_k(F(x_k) - F(x_*)) + \tfrac{1+\mu A_k}{2}\|z_k - x_*\|^2 + \tfrac{A_{k+1}}{2}\xi_k\end{aligned}$$

$$
\begin{aligned}
&- \tfrac{A_k(A_{k+1}-A_k)\mu(1+A_k\mu)}{2(A_{k+1}+A_k(2A_{k+1}-A_k)\mu)}\|x_k - z_k\|^2 \\
&- \tfrac{A_{k+1}\lambda_k}{2(1-\sigma_k^2)}\|\nabla f(x_{k+1}) - \nabla f(y_k) + (1-\sigma_k^2)\tfrac{y_k - x_{k+1}}{\lambda_k}\|^2 \\
&- \tfrac{\mu(A_{k+1}+A_k(2A_{k+1}-A_k)\mu)}{2(1+A_{k+1}\mu)}\|x_{k+1} - y_k + \tfrac{(A_{k+1}-A_k)^2}{A_{k+1}+A_k(2A_{k+1}-A_k)\mu}(v_{k+1} + \nabla f(y_k))\|^2 \\
&+ A_{k+1}\tfrac{A_{k+1}(A_{k+1}-\eta_k)-A_k(1+\eta_k\mu)(2A_{k+1}-A_k)}{2(A_{k+1}+A_k(2A_{k+1}-A_k)\mu)}\|v_{k+1} + \nabla f(y_k)\|^2 \\
&\leq A_k(F(x_k) - F(x_*)) + \tfrac{1+\mu A_k}{2}\|z_k - x_*\|^2 + \tfrac{A_{k+1}}{2}\xi_k \\
&+ A_{k+1}\tfrac{A_{k+1}(A_{k+1}-\eta_k)-A_k(1+\eta_k\mu)(2A_{k+1}-A_k)}{2(A_{k+1}+A_k(2A_{k+1}-A_k)\mu)}\|v_{k+1} + \nabla f(y_k)\|^2 \\
&= A_k(F(x_k) - F(x_*)) + \tfrac{1+\mu A_k}{2}\|z_k - x_*\|^2 + \tfrac{A_{k+1}}{2}\xi_k,
\end{aligned}
$$

where the inequality in the second to last line comes from the fact that factors in front of three squared Euclidean norms are nonpositive. In addition, the last equality follows from the particular choice of $A_{k+1}$ satisfies

$$A_{k+1}(A_{k+1} - \eta_k) - A_k(1 + \eta_k\mu)(2A_{k+1} - A_k) = 0,$$

which implies that the factors in front of the last squared Euclidean norm vanishes. ■

## 6.4 Numerical examples

In this section, we present a few numerical experiments illustrating the behavior of the accelerated inexact forward backward method (Algorithm 6.3.1) on two convex problems. More precisely, we applied the method to a factorization problem and to a total variation problem.

In both cases, we use a Tikhonov regularization, improving the conditioning and rendering the problems strongly convex, and illustrate the numerical performances of the algorithm with different tunings, including in the purely relative ($\xi_k = 0$) and absolute accuracy ($\sigma_k = \zeta_k = 0$) setups, as well as the influence of the knowledge of strong convexity parameter.

### 6.4.1 Factorization problem

Our first numerical experiment is a CUR-like factorization problem, introduced in Mairal et al. [2011]. It consists, given a matrix $W \in \mathrm{R}^{m\times p}$, in solving the minimization problem

$$\min_X F(X) \equiv \underbrace{\tfrac{1}{2}\|W - WXW\|_F^2}_{f(X)} + \underbrace{\lambda_{\text{row}}\sum_{i=1}^{n_r}\|X^i\|_2 + \lambda_{\text{col}}\sum_{j=1}^{n_c}\|X_j\|_2 + \tfrac{\mu_{\text{reg}}}{2}\|X\|_F^2}_{g(X)},$$

where $\|\cdot\|_F$ is the Frobenius norm, and where $X^i$ and $X_j$ respectively denote the $i$th row and the $j$th column of the matrix $X$. This problem has already been used in Schmidt et al. [2011] for illustrating convergence guarantees of an inexact accelerated proximal gradient method with absolute errors. As in Schmidt et al. [2011], we use an inexact version of the proximal operator of the regularization part, which we solve via a dual block coordinate ascent method [Jenatton et al., 2010] (i.e., we solve the dual of the proximal problem). Our implementation (see link in §Codes from Section 6.1) is based on that of Schmidt et al. [2011], and our experiments are done on the "a1a" dataset from the LIBSVM library [Chang and Lin, 2011]. The corresponding matrix $W$ is normalized for having zero mean and unit norm. We also impose $\lambda_{\text{col}} = \sqrt{\frac{p}{m}}\lambda_{\text{row}}$ for having a similar scaling for the row and column

regularization parameters. The choice of the error criteria and regularization parameters is detailed in Figure 6.4.1 where we plot gaps between the objective function values at the iterates of Algorithm 6.3.1 and the optimal objective value versus the number of iteration of Algorithm 6.3.1 (left) and versus the total number of dual block coordinate ascent iterations (right).

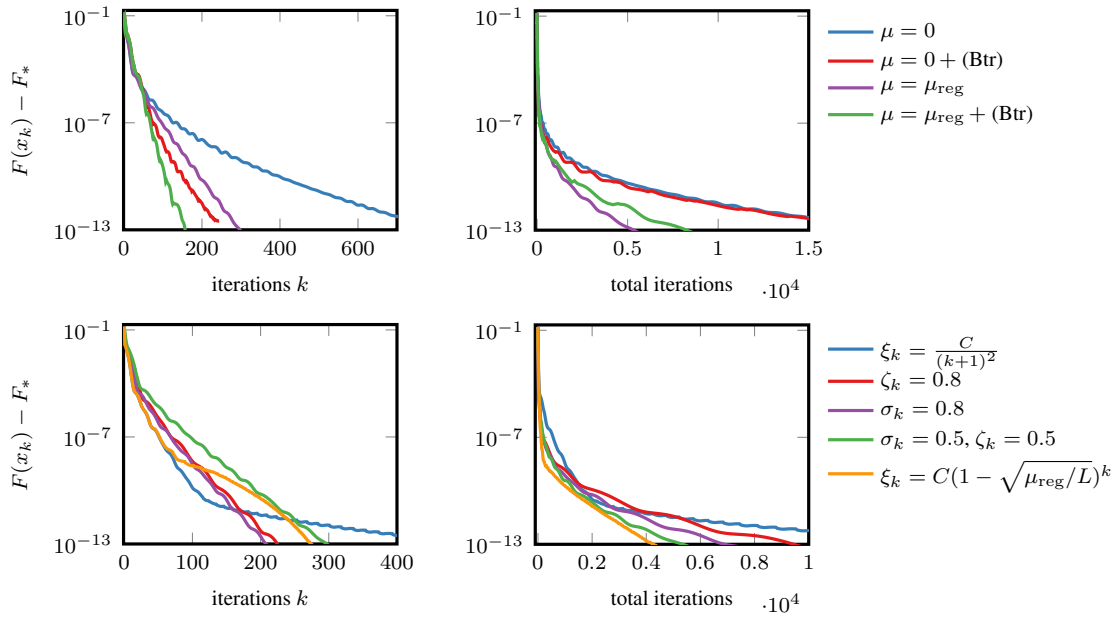


Figure 6.4.1: Algorithm 6.3.1 on CUR factorization. The initial step size is set to $\lambda_0 = \frac{1-\sigma_0^2}{L}$, initial $L$ to $\|W\|^4$, $\lambda_{\text{row}} = \lambda_{\text{col}}\sqrt{m/p} = 2.10^{-3}$ ($\sim 30\%$ nonzero coefficients in the solution) and $\mu_{\text{reg}} = 2.10^{-3}L$. Top: $\sigma_k = 0.8$ , $\zeta_k = 0$ and $\xi_k = 0$. Bottom: accelerated inexact forward-backward with $\mu = \mu_{\text{reg}}$ and no backtracking. When backtracking is used, $\alpha$ is set to $\frac{1}{2}$ and $\beta$ to 1.1. "Total iterations" refers to total the number of block coordinate ascent iterations used in the subroutine that computes the proximal steps approximately. $F_*$ is approximated by the smallest objective values encountered in $2.10^4$ total iterations of block coordinate ascent.

### 6.4.2 Total variation regularization

In this section, we compare the behaviors of the accelerated inexact forward backward method (Algorithm 6.3.1) with different tunings, on the classical problem of deblurring through total variation regularization [Rudin et al., 1992, Rudin and Osher, 1994, Wang et al., 2008]. Given a blurred image $Y \in \mathrm{R}^{n\times n}$ and a blurring operator $A$, the problem consists in solving

$$\min_X F(X) \equiv \underbrace{\tfrac{1}{2}\|AX - Y\|_F^2}_{f(X)} + \underbrace{\lambda_{\text{reg}} \sum_{i,j=0}^{n} \|(\nabla X)_{i,j}\|_2 + \tfrac{\mu_{\text{reg}}}{2}\|X\|_F^2}_{g(X)},$$

where $\nabla$ is the discrete gradient of an image, see e.g., Chambolle and Pock [2016, Equation (2.4)]. One way of dealing with this problem is to approximate the proximal operator of the discrete total variation plus the Tikhonov regularization. As in Villa et al. [2013], Millán and Machado [2019], we apply FISTA [Beck and Teboulle, 2009] on the dual of the proximal subproblem (which is provided

e.g., in Chambolle and Pock [2016, Example 3.1]), which we use in the accelerated inexact forward-backward method.

In the experiments $Y$ is the popular $256 \times 256$ greyscale boat image (see e.g., `http://sipi.usc.edu/database/`). We blur $Y$ via a $5 \times 5$ box blur kernel $A$, and add a Gaussian noise of standard deviation $0.01$ times the mean of the blurred image and zero mean to the picture. Some results are detailed in Figure 6.4.2 where we plot gaps between the objective function values at the iterates of Algorithm 6.3.1 and the optimal objective value versus the number of iterations of Algorithm 6.3.1 (left) and versus the total number iterations of FISTA on the dual subproblem (right).

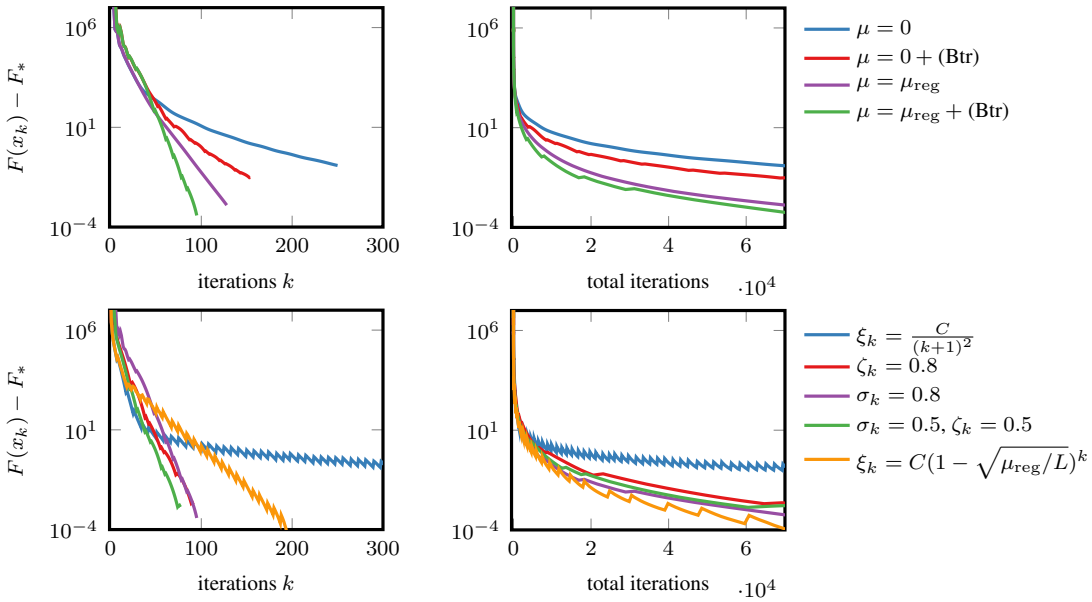


Figure 6.4.2: Algorithm 6.3.1 on TV regularization. The initial step size is set to $\lambda_0 = \frac{1-\sigma_0^2}{L}$, the initial $L$ and $\lambda_{\text{reg}}$ to 1 and $\mu_{reg}$ to $10^{-2}L$. Top: $\sigma_k = 0.8$ , $\zeta_k = 0$ and $\xi_k = 0$. Bottom: accelerated inexact forward-backward with $\mu = \mu_{\text{reg}}$ and backtracking. When backtracking is used, $\alpha$ is set to $\frac{1}{2}$ and $\beta$ to 1.1. "Total iterations" refers to total the number of FISTA iterations used in the subroutine that computes the proximal steps approximately. $F_*$ is approximated by the smallest objective values encountered in $2.10^4$ total FISTA iterations.

## 6.5 An accelerated hybrid proximal extragradient method

In this section, we provide an improved analysis for the specific case $f = 0$ (no smooth convex term in (6.1)). This type of methods is often used as a *globalization* strategy for higher-order methods, see Monteiro and Svaiter [2013]. The version presented in this section allows exploiting the possible strong convexity of the objective, which was not incorporated in previous versions of the method, to the best of our knowledge.

### 6.5.1 Algorithm

A version of Algorithm 6.5.1 has been re-analyzed and used by Alves [2021] recently for accelerating higher-order tensor algorithms.

When $\mu = 0$ and $\sigma_k$ is fixed, this method actually reduces to the optimized relatively inexact proximal point algorithm (ORI-PPA) from Chapter 5. In this case, the growth rate of the sequence $\{A_k\}_k$ is essentially

$$A_k \geq \frac{1}{4}\left(\sum_{i=0}^{k-1}\sqrt{\frac{2\lambda_i}{(1+\sigma_i)}}\right)^2 \geq O(k^2) \quad \text{for } k \geq 1,$$

when the parameters $\{\lambda_k\}_k$, $\{\sigma_k\}_k$ are well chosen (e.g., constant parameters).

**Strongly convex accelerated hybrid proximal extragradient method** (Algorithm 6.5.1)

**Input:**

- Objective function: $g \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$.
- Initial point: $x_0 \in \mathbb{R}^d$.
- Step sizes: $\{\lambda_k\}_k$ with $\lambda_k > 0$.
- Tolerance parameters: $\{\sigma_k\}_k$ with $\sigma_k \in [0, 1]$.

**Initialization:**

$z_0 = x_0$, $A_0 = 0$,

**Run:**

For $k = 0, 1, \ldots$:

$$\begin{aligned}
a_{k+1} &= \frac{(2(1-\sigma_k)+\lambda_k\mu)\lambda_k\left(1+2A_k\mu+\sqrt{1+4A_k(1+A_k\mu)\frac{(1+\lambda_k\mu)^2-\sigma_k(\sigma_k+\lambda_k\mu)}{(2(1-\sigma_k)+\lambda_k\mu)\lambda_k}}\right)}{2(1-\sigma_k^2+\lambda_k\mu\sigma_k)} \\
A_{k+1} &= A_k + a_{k+1} \\
y_k &= x_k + \frac{(A_{k+1}-A_k)(1+\mu A_k)}{A_{k+1}+\mu A_k(2A_{k+1}-A_k)}(z_k - x_k) \\
\varepsilon_k &= \frac{\sigma_k^2}{2(1+\lambda_k\mu)^2}\|x_{k+1} - y_k\|^2 \\
(x_{k+1},\ v_{k+1}) &\approx_{\varepsilon_k,\mu} \left(\mathrm{prox}_{\lambda_k g}(y_k), \mathrm{prox}_{g^*/\lambda_k}(\tfrac{y_k}{\lambda_k})\right) \\
z_{k+1} &= z_k + \frac{A_{k+1}-A_k}{1+\mu A_{k+1}}\left(\mu(x_{k+1} - z_k) - v_{k+1}\right)
\end{aligned}$$

End For

**Output:** $x_{k+1}$

When $\mu > 0$, the sequence $\{A_k\}_k$ grows as

$$\begin{aligned}
A_{k+1} &\geq A_k\left(1 + \frac{2(1-\sigma_k)+\lambda_k\mu}{1-\sigma_k^2+\lambda_k\mu\sigma_k}\left(\lambda_k\mu + \sqrt{\lambda_k\mu\frac{(1+\lambda_k\mu)^2-\sigma_k(\sigma_k+\lambda_k\mu)}{(2(1-\sigma_k)+\lambda_k\mu)}}\right)\right) \\
&= A_k\left(\frac{(1+\lambda_k\mu)^2-\sigma_k(\sigma_k+\lambda_k\mu)+\sqrt{\lambda_k\mu(2(1-\sigma_k)+\lambda_k\mu)((1+\lambda_k\mu)^2-\sigma_k(\sigma_k+\lambda_k\mu))}}{1-\sigma_k^2+\lambda_k\mu\sigma_k}\right)
\end{aligned}$$

$$= A_k \left( \frac{(1+\lambda_k\mu)^2 - \sigma_k(\sigma_k+\lambda_k\mu) - \lambda_k\mu(2(1-\sigma_k)+\lambda_k\mu)}{1-\sigma_k^2+\lambda_k\mu\sigma_k} \right) / \left( 1 - \sqrt{\frac{\lambda_k\mu(2(1-\sigma_k)+\lambda_k\mu)}{(1+\lambda_k\mu)^2-\sigma_k(\sigma_k+\lambda_k\mu)}} \right)$$
$$= A_k / \left( 1 - \sqrt{\frac{\lambda_k\mu(2(1-\sigma_k)+\lambda_k\mu)}{(1+\lambda_k\mu)^2-\sigma_k(\sigma_k+\lambda_k\mu)}} \right),$$

with $A_1 = \lambda_0 \frac{2(1-\sigma_0)+\lambda_0\mu}{(1-\sigma_0^2+\lambda_0\mu\sigma_0)} \geq \lambda_0$. In particular we recover the rate of the inexact accelerated forward-backward method when $\sigma_k = 1$. In addition, we notice that

$$1 - \sqrt{\frac{\lambda_k\mu(2+\lambda_k\mu-2\sigma_k)}{(1+\lambda_k\mu)^2-\sigma_k(\sigma_k+\lambda_k\mu)}} \sim 1 - \sqrt{\frac{2}{1+\sigma_k}\lambda_k\mu},$$

when $\lambda_k\mu \ll 0$.

**Theorem 6.5.1.** *Let $g \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$, $k \geq 0$, a parameter $\sigma_k \in [0,1]$ and some $\lambda_k > 0$. For any $x_k, z_k \in \mathbb{R}^d$ and $A_k \geq 0$, it holds that*

$$A_{k+1}(g(x_{k+1}) - g(x_*)) + \frac{1+\mu A_{k+1}}{2}\|z_{k+1} - x_*\|^2 \leq A_k(g(x_k) - g(x_*)) + \frac{1+\mu A_k}{2}\|z_k - x_*\|^2$$

*with $x_* \in \operatorname{argmin}_x g(x)$, and where $z_{k+1}$, $x_{k+1}$ are constructed by one iteration of Algorithm 6.5.1.*

*Proof.* The proof of this Theorem is deferred to Section 6.5.2 ■

Just as for the its forward-backward version, one can obtain a final worst-case guarantee driven by the growth rate of the sequence $\{A_k\}_k$.

**Corollary 6.5.2.** *Let $g \in \mathcal{F}_{\mu,\infty}(\mathbb{R}^d)$, $\{\lambda_k\}_k$ be a sequence of positive parameters, and a sequence (relative error parameters) $\{\sigma_k\}_k$ satisfying $\sigma_k \in [0,1]$. Let $x_N \in \mathbb{R}^d$ be the output after $N \in \mathbb{N}^*$ iterations of Algorithm 6.5.1 on $g$ initiated at $x_0 \in \mathbb{R}^d$, it holds that*

$$g(x_N) - g(x_*) \leq \frac{1}{2A_N}\|x_0 - x_*\|^2,$$

*where $x_* \in \operatorname{argmin}_x g(x)$.*

*Proof.* The proof follows from the same lines as that of Corollary 6.3.3, using Theorem 6.5.1 instead of Theorem 6.3.2. ■

### 6.5.2 Proof of Theorem 6.5.1

The proof follows the same structure as that of in Section 6.3.2, and simply consists in reformulating a weighted sum of inequalities.

*Proof.* First we consider $\sigma_k \in (0,1]$ as the case $\sigma_k = 0$ requires a particular treatment. Let $w_{k+1} \in \mathbb{R}^d$ such that $v_{k+1} - \mu x_{k+1} + \mu w_{k+1} \in \partial g(w_{k+1})$. Using (6.3), this leads to

$$\begin{aligned}\mathrm{PD}_{\frac{\lambda_k}{1+\mu\lambda_k} g_\mu}(x_{k+1}, v_{k+1} - \mu x_{k+1}; \tfrac{y_k}{1+\mu\lambda_k}) = \tfrac{1}{2(1+\lambda_k\mu)^2}\|x_{k+1} - y_k + \lambda_k v_{k+1}\|^2 \\ + \tfrac{\lambda_k}{1+\lambda_k\mu}\Big(g(x_{k+1}) - g(w_{k+1}) + \tfrac{\mu}{2}\|x_{k+1} - w_{k+1}\|^2 \\ - \langle x_{k+1} - w_{k+1}, v_{k+1}\rangle\Big),\end{aligned}$$

with $g_\mu(\cdot) = g(\cdot) - \frac{\mu}{2}\|\cdot\|^2$.

The proof consists in performing a weighted sum of the following inequalities:

- strong convexity between $w_{k+1}$ and $x_*$ with weight $\nu_1 = A_{k+1} - A_k$

$$g(x_*) \geq g(w_{k+1}) + \langle v_{k+1} - \mu x_{k+1} + \mu w_{k+1}, x_* - w_{k+1}\rangle + \tfrac{\mu}{2}\|w_{k+1} - x_*\|^2,$$

- strong convexity between $w_{k+1}$ and $x_k$ with weight $\nu_2 = A_k$

$$g(x_k) \geq g(w_{k+1}) + \langle v_{k+1} - \mu x_{k+1} + \mu w_{k+1}, x_k - w_{k+1}\rangle + \tfrac{\mu}{2}\|w_{k+1} - x_k\|^2,$$

- strong convexity between $w_{k+1}$ and $x_{k+1}$ with weight $\nu_3 = \frac{A_{k+1}(1-\sigma_k+\lambda_k\mu)}{\sigma_k}$

$$g(x_{k+1}) \geq g(w_{k+1}) + \langle v_{k+1} - \mu x_{k+1} + \mu w_{k+1}, x_{k+1} - w_{k+1}\rangle + \tfrac{\mu}{2}\|w_{k+1} - x_{k+1}\|^2,$$

- approximation requirement on $x_{k+1}$ with weight $\nu_4 = \frac{A_{k+1}}{\lambda_k\sigma_k}$

$$\begin{aligned}\tfrac{\sigma_k^2}{2}\|x_{k+1} - y_k\|^2 \geq \tfrac{1}{2}\|x_{k+1} - y_k + \lambda_k v_{k+1}\|^2 + \lambda_k(1+\lambda_k\mu)\Big(g(x_{k+1}) - g(w_{k+1}) \\ + \tfrac{\mu}{2}\|x_{k+1} - w_{k+1}\|^2 - \langle x_{k+1} - w_{k+1}, v_{k+1}\rangle\Big).\end{aligned}$$

Substituting $y_k$ and $z_{k+1}$ in the weighted sum, that is

$$\begin{aligned} y_k &= x_k + \tfrac{(A_{k+1}-A_k)(1+\mu A_k)}{A_{k+1}+\mu A_k(2A_{k+1}-A_k)}(z_k - x_k) \\ z_{k+1} &= z_k + \tfrac{A_{k+1}-A_k}{1+\mu A_{k+1}}\left(\mu(x_{k+1} - z_k) - v_{k+1}\right), \end{aligned}$$

the weighted sum is equivalently reformulated as

$$\begin{aligned}
&A_{k+1}(g(x_{k+1}) - g(x_*)) + \tfrac{1+\mu A_{k+1}}{2}\|z_{k+1} - x_*\|^2 \\
&\leq A_k(g(x_k) - g(x_*)) + \tfrac{1+\mu A_k}{2}\|z_k - x_*\|^2 \\
&\quad - \tfrac{A_k(A_{k+1}-A_k)\mu(1+A_k\mu)}{2(A_{k+1}+A_k(2A_{k+1}-A_k)\mu)}\|x_k - z_k\|^2 \\
&\quad - \tfrac{\lambda_k}{2}\tfrac{\lambda_k\mu(A_{k+1}+A_k(2A_{k+1}-A_k)\mu)}{1+A_{k+1}\mu}\|\tfrac{y_k - x_{k+1}}{\lambda_k} - \tfrac{\left(\mu(A_{k+1}^2+A_k^2\sigma_k)+A_{k+1}(1-\sigma_k-2A_k\mu\sigma_k)\right)v_{k+1}}{A_{k+1}(1+A_{k+1\mu})(1-\sigma_k^2)+\lambda_k\mu\sigma_k(A_{k+1}+A_k(2A_{k+1}-A_k)\mu)}\|^2 \\
&\quad - \tfrac{\lambda_k(1-\sigma_k^2)A_{k+1}}{2\sigma_k}\|\tfrac{y_k - x_{k+1}}{\lambda_k} - \tfrac{\mu(A_{k+1}^2+A_k^2\sigma_k)+A_{k+1}(1-\sigma_k-2A_k\mu\sigma_k)}{A_{k+1}(1+A_{k+1\mu})(1-\sigma_k^2)+\lambda_k\mu\sigma_k(A_{k+1}+A_k(2A_{k+1}-A_k)\mu)}v_{k+1}\|^2 \\
&\quad + \tfrac{A_{k+1}((2A_kA_{k+1}-A_k^2)(1-\sigma_k^2+\lambda_k^2\mu^2+\lambda_k\mu(2-\sigma_k))-A_{k+1}^2(1-\sigma_k^2+\lambda_k\mu\sigma_k)+A_{k+1}\lambda_k(2(1-\sigma_k)+\lambda_k\mu))}{2(A_k^2\lambda_k\mu^2\sigma_k-A_{k+1}^2\mu(1-\sigma_k^2)-A_{k+1}(1-\sigma_k^2+\lambda_k\mu\sigma_k(1+2A_k\mu)))}\|v_{k+1}\|^2 \\
&\leq A_k(g(x_k) - g(x_*)) + \tfrac{1+\mu A_k}{2}\|z_k - x_*\|^2 \\
&\quad + \tfrac{A_{k+1}((2A_kA_{k+1}-A_k^2)(1-\sigma_k^2+\lambda_k^2\mu^2+\lambda_k\mu(2-\sigma_k))-A_{k+1}^2(1-\sigma_k^2+\lambda_k\mu\sigma_k)+A_{k+1}\lambda_k(2(1-\sigma_k)+\lambda_k\mu))}{2(A_k^2\lambda_k\mu^2\sigma_k-A_{k+1}^2\mu(1-\sigma_k^2)-A_{k+1}(1-\sigma_k^2+\lambda_k\mu\sigma_k(1+2A_k\mu)))}\|v_{k+1}\|^2 \\
&= A_k(g(x_k) - g(x_*)) + \tfrac{1+\mu A_k}{2}\|z_k - x_*\|^2,
\end{aligned}$$

where the inequality in the second to last line comes from the fact that factors in front of squared Euclidean norms are nonpositive, and the last equality from the fact that $A_{k+1}$ is chosen such that it satisfies

$$(2A_kA_{k+1}-A_k^2)(1-\sigma_k^2+\lambda_k^2\mu^2+\lambda_k\mu(2-\sigma_k))-A_{k+1}^2(1-\sigma_k^2+\lambda_k\mu\sigma_k)+A_{k+1}\lambda_k(2(1-\sigma_k)+\lambda_k\mu) = 0.$$

Note that the intermediary expressions largely simplifies when choosing this $A_{k+1}$, as the last term disappears, and the two other squared Euclidean norms become (up to nonpositive multiplicative factors) $\|x_k - z_k\|^2$ and $\|\frac{y_k - x_{k+1}}{\lambda_k} - \frac{1-\sigma_k+\lambda_k\mu}{1-\sigma_k^2+\lambda_k\mu}v_{k+1}\|^2$.

For the case $\sigma_k = 0$ (i.e., exact proximal computations) $v_{k+1} \in \partial g(x_{k+1})$ and we proceed as previously by performing the following weighted sum of inequalities:

- strong convexity between $x_{k+1}$ and $x_*$ with weight $\nu_1 = A_{k+1} - A_k$

$$g(x_*) \geq g(x_{k+1}) + \langle v_{k+1}, x_* - x_{k+1}\rangle + \tfrac{\mu}{2}\|x_{k+1} - x_*\|^2,$$

- strong convexity between $x_{k+1}$ and $x_k$ with weight $\nu_2 = A_k$

$$g(x_k) \geq g(x_{k+1}) + \langle v_{k+1}, x_k - x_{k+1}\rangle + \tfrac{\mu}{2}\|x_{k+1} - x_k\|^2.$$

Substituting $y_k$, $x_{k+1}$ and $z_k$ in the weighted sum, that is

$$\begin{aligned} y_k &= x_k + \tfrac{(A_{k+1}-A_k)(1+\mu A_k)}{A_{k+1}+\mu A_k(2A_{k+1}-A_k)}(z_k - x_k) \\ x_{k+1} &= y_k - \lambda_k v_{k+1} \\ z_{k+1} &= z_k + \tfrac{A_{k+1}-A_k}{1+\mu A_{k+1}}\left(\mu(x_{k+1} - z_k) - v_{k+1}\right), \end{aligned}$$

the weighted sum is equivalently reformulated as

$$\begin{aligned} &A_{k+1}(g(x_{k+1}) - g(x_*)) + \tfrac{1+\mu A_{k+1}}{2}\|z_{k+1} - x_*\|^2 \\ &\leq A_k(g(x_k) - g(x_*)) + \tfrac{1+\mu A_k}{2}\|z_k - x_*\|^2 \\ &\quad - \tfrac{A_k(A_{k+1}-A_k)\mu(1+A_k\mu)}{2(A_{k+1}+A_k(2A_{k+1}-A_k)\mu)}\|x_k - z_k\|^2 \\ &\quad + \tfrac{A_{k+1}(A_{k+1}-\lambda_k(2+\lambda_k\mu))-A_k(2A_{k+1}-A_k)(1+\lambda_k\mu)^2}{2(1+A_{k+1}\mu)}\|v_{k+1}\|^2 \\ &\leq A_k(g(x_k) - g(x_*)) + \tfrac{1+\mu A_k}{2}\|z_k - x_*\|^2 \\ &\quad + \tfrac{A_{k+1}(A_{k+1}-\lambda_k(2+\lambda_k\mu))-A_k(2A_{k+1}-A_k)(1+\lambda_k\mu)^2}{2(1+A_{k+1}\mu)}\|v_{k+1}\|^2 \\ &= A_k(g(x_k) - g(x_*)) + \tfrac{1+\mu A_k}{2}\|z_k - x_*\|^2, \end{aligned}$$

where the inequality in the second to last line comes from the fact that factor in front of squared Euclidean norm is nonpositive, and the last equality from the fact that $A_{k+1}$ is chosen such that it satisfies

$$A_{k+1}(A_{k+1} - \lambda_k(2 + \lambda_k\mu)) - A_k(2A_{k+1} - A_k)(1 + \lambda_k\mu)^2 = 0,$$

when $\sigma_k = 0$. ■

## 6.6 Partially inexact Douglas-Rachford splitting algorithm

In this section, we study a partially inexact version of the Douglas-Rachford algorithm introduced in Eckstein and Yao [2018, Algorithm 3]. In Eckstein and Yao [2018, Proposition 3], convergence of the scheme is established for solving monotone inclusions problems. We study the particular case that consists in applying this scheme to the composite convex optimization problem of the form

$$\min_x \{F(x) \equiv f(x) + g(x)\},$$

where $f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$ and $g \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$. In the case where exact proximal operations are used, we recover the convergence rates of the vanilla Douglas-Rachford splitting from Giselsson and Boyd [2016, Theorem 2] (vanilla DRS is obtained by fixing $\alpha = 1/2$ in that work) and Giselsson and Boyd [2014, Proposition 6] (vanilla DRS is obtained by fixing $\theta = 1/2$ in that work). The proofs below were, again, obtained through the same methodology.

This algorithm uses an inexactness criterion that is slightly more restrictive than the previous ones. Indeed, here the iterate $x_{k+1}$ is an approximated proximal step from $y_k$, denoted $x_{k+1} \approx_\sigma \operatorname{prox}_{\lambda f}(y_k)$, if it satisfies

$$\|x_k - z_k + \lambda \nabla f(x_k)\|^2 \leq \sigma^2 \|y_k - z_k + \lambda \nabla f(x_k)\|^2. \tag{6.9}$$

**Partially inexact Douglas-Rachford splitting algorithm [Eckstein and Yao, 2018]**

**Input:**

- Objective function: $f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$, $g \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$.
- Initial point: $z_0 \in \mathbb{R}^d$.
- Step size: $\lambda > 0$.
- Tolerance parameter: $\sigma \in [0, 1)$.

**Run:**

For $k = 0, 1, \ldots$:

$$\begin{aligned} x_k &\approx_\sigma \operatorname{prox}_{\lambda f}(z_k) \\ y_k &= \operatorname{prox}_{\lambda g}(x_k - \lambda \nabla f(x_k)) \\ z_{k+1} &= z_k + y_k - x_k \end{aligned}$$

End For

**Output:** $x_k$

**Theorem 6.6.1.** *Let $f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$ and $g \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$, $F \equiv f + g$. Let $\lambda > 0$ and $\sigma \in [0, 1)$. For any $z_k \in \mathbb{R}^d$, it holds that*

$$\|z_{k+1} - z_*\| \leq \max\left(\frac{1 - \sigma + \lambda\mu\sigma}{1 - \sigma + \lambda\mu}, \frac{\sigma + (1-\sigma)\lambda L}{1 + (1-\sigma)\lambda L}\right) \|z_k - z_*\|$$

*with $z_*$ being such that $\operatorname{prox}_{\lambda f}(z_*) = x_*$ with $x_* \in \operatorname{argmin}_x F(x)$, and where $z_{k+1}$ is constructed by one iteration of the Partially inexact Douglas-Rachford splitting algorithm. Furthermore, this bound is tight: for all $\lambda > 0$, $\sigma \in [0, 1)$, $d \in \mathbb{N}^*$ and $z_k \in \mathbb{R}^d$, there exists $f \in \mathcal{F}_{\mu,\mathrm{L}}(\mathbb{R}^d)$ and $g \in \mathcal{F}_{0,\infty}(\mathbb{R}^d)$ such that this bound is achieved with equality for all $k \in \mathbb{N}$.*

*Proof.* Let us denote $\rho(\lambda, \sigma) := \max\left(\frac{1-\sigma+\lambda\mu\sigma}{1-\sigma+\lambda\mu}, \frac{\sigma+(1-\sigma)\lambda L}{1+(1-\sigma)\lambda L}\right)$ (which we note $\rho$ when values of $\lambda$ and $\sigma$ are clear from the context), and remark that

$$\rho(\lambda, \sigma) = \begin{cases} \frac{1-\sigma+\lambda\mu\sigma}{1-\sigma+\lambda\mu} & \text{if } \lambda \leq \frac{1}{\sqrt{L\mu}}, \\ \frac{\sigma+(1-\sigma)\lambda L}{1+(1-\sigma)\lambda L} & \text{otherwise.} \end{cases}$$

The proof consists, as before, in performing the following weighted sum of inequalities.

- Strong convexity and smoothness of $f$ between $x_k$ and $x_*$ with $\nu_1 = 2\lambda\rho(\lambda,\sigma)$

$$\begin{aligned}&\langle \nabla f(x_k) - \nabla f(x_*), x_k - x_*\rangle\\ &\qquad \geq \tfrac{1}{L}\|\nabla f(x_k) - \nabla f(x_*)\|^2 + \tfrac{\mu}{1-\frac{\mu}{L}}\|x_k - x_* - \tfrac{1}{L}(\nabla f(x_k) - \nabla f(x_*))\|^2,\end{aligned}$$

  see e.g., [Taylor et al., 2017c, Theorem 4] (symmetrized version).

- Convexity of $g$ between $y_k$ and $x_*$ with weight $\nu_2 = 2\lambda\rho(\lambda,\sigma)\eta(\lambda)$

$$\langle s_g(y_k) - s_g(x_*), y_k - x_*\rangle \geq 0,$$

  for some $s_g(y_k) \in \partial g(y_k)$, and $s_g(x_*) \in \partial g(x_*)$, and $\eta(\lambda) := \min(\frac{1+\lambda\mu}{1-\lambda\mu}, \frac{1+\lambda L}{\lambda L-1})$.

- Accuracy requirement (6.9) with weight $\nu_3 = \frac{\rho(\lambda,\sigma)}{\sigma}$

$$\|x_k - z_k + \lambda\nabla f(x_k)\|^2 \leq \sigma^2\|z_k - y_k - \lambda\nabla f(x_k)\|^2.$$

Substituting $x_k$,$y_k$, $z_{k+1}$, $x_*$, and $s_g(x_*)$ by the expressions

$$\begin{aligned}y_k &= x_k - \lambda\nabla f(x_k) - \lambda s_g(y_k)\\ z_{k+1} &= z_k - \lambda\nabla f(x_k) - \lambda s_g(y_k)\\ x_* &= z_* - \lambda\nabla f(x_*)\\ s_g(x_*) &= -\nabla f(x_*) \text{ (optimality conditions for } x_*),\end{aligned}$$

we then reformulate the weighted sum as follows, in two cases.

- When $\lambda \leq \frac{1}{\sqrt{L\mu}}$, we have $\rho = \frac{1-\sigma+\lambda\mu\sigma}{1-\sigma+\lambda\mu}$, $\eta(\lambda) = \frac{1+\lambda\mu}{1-\lambda\mu}$, and

$$\begin{aligned}&0 \leq \rho^2\|z_k - z_*\|^2 - \|z_{k+1} - z_*\|^2\\ &-\rho\frac{2\lambda(1-\lambda^2\mu L)}{(1-\lambda^2\mu^2)(L-\mu)}\|\nabla f(x_*) - \nabla f(x_k) + \mu(x_k - z_* + \lambda\nabla f(x_*))\|^2\\ &-\rho\frac{\lambda^2(\lambda\mu(\sigma+1)-\sigma+1)^2}{(1-\lambda^2\mu^2)\,(\lambda\mu(\sigma^2+1)-\sigma^2+1)}\|\nabla f(x_k) + (\lambda\mu+1)s_g(y_k) + \mu(z_* - x_k)\|^2\\ &-\frac{\rho-\sigma}{\sigma(\rho\sigma+1)}\left\|\frac{\lambda\sigma(1-\rho\sigma)}{(\rho-\sigma)}s_g(y_k) - \lambda\nabla f(x_k) + \frac{\rho(\sigma^2-1)}{(\rho-\sigma)}x_k + (\rho\sigma+1)z_k - \frac{(\rho^2-1)\sigma}{(\rho-\sigma)}z_*\right\|^2,\end{aligned}$$

  and the conclusion $\|z_{k+1} - z_*\|^2 \leq \rho^2\|z_k - z_*\|^2$ follows from the signs of all leading coefficients being nonnegative, which easily follows from $\lambda \leq 1/\sqrt{L\mu}$, $\lambda > 0$, $\rho > 0$, $L > \mu > 0$, $0 \leq \sigma < 1$, and $\rho - \sigma = (1-\sigma)^2/(\lambda\mu - \sigma + 1) > 0$.

- When $\lambda \geq \frac{1}{\sqrt{L\mu}}$, we have $\rho = \frac{\sigma+(1-\sigma)\lambda L}{1+(1-\sigma)\lambda L}$, $\eta(\lambda) = \frac{1+\lambda L}{\lambda L-1}$, and

$$\begin{aligned}&0 \leq \rho^2\|z_k - z_*\|^2 - \|z_{k+1} - z_*\|^2\\ &-\rho\frac{2\lambda(\lambda^2\mu L-1)}{(\lambda^2L^2-1)\,(L-\mu)}\|\nabla f(x_*) - \nabla f(x_k) + L(x_k - z_* + \lambda\nabla f(x_*))\|^2\\ &-\rho\frac{(L\lambda(1-\sigma)+\sigma+1)^2}{\lambda L\,(\lambda^2L^2-1)\,(\lambda L\,(1-\sigma^2)+\sigma^2+1)}\|\lambda(\lambda L\nabla f(x_k) + (\lambda L+1)s_g(y_k)) - x_k + z_*\|^2\\ &-\frac{\rho-\sigma}{\sigma(\rho\sigma+1)}\left\|\frac{\lambda\sigma(1-\rho\sigma)}{(\rho-\sigma)}s_g(y_k) - \lambda\nabla f(x_k) + \frac{\rho\left(\sigma^2-1\right)}{(\rho-\sigma)}x_k + (\rho\sigma+1)z_k - \frac{\left(\rho^2-1\right)\sigma}{(\rho-\sigma)}z_*\right\|^2,\end{aligned}$$

and the desired conclusion $\|z_{k+1} - z_*\|^2 \leq \rho^2 \|z_k - z_*\|^2$ follows from leading coefficients being nonnegative, using $\lambda \geq 1/\sqrt{L\mu}$, $\lambda > 0$, $\rho > 0$, $L > \mu > 0$, $0 \leq \sigma < 1$, and $\rho - \sigma = \lambda L(1-\sigma)^2/(1 + \lambda L(1-\sigma)) > 0$.

Tightness relies on applying the method on either the pair $f(x) = \frac{\mu}{2}x^2$ and $g(x) = 0$, or on the pair $f(x) = \frac{L}{2}x^2$ and $g(x) = i_{\{0\}}(x)$ (the indicator function of 0), just as in the exact case ($\sigma = 0$), see Giselsson and Boyd [2016, Section 3.2].

- the first term in the maximum is achieved by picking $f(x) = \frac{\mu}{2}x^2$ and $g(x) = 0$. In this case the solution is $x_* = z_* = 0$, and picking $x_k = \frac{1-\sigma}{1-\sigma+\lambda\mu} z_k$, $y_k = x_k - \lambda \nabla f(x_k) = \frac{(1-\sigma)(1-\lambda\mu)}{1-\sigma+\lambda\mu} z_k$, and $z_{k+1} = \frac{1-\sigma+\lambda\mu\sigma}{1-\sigma+\lambda\mu} z_k$ is a valid sequence for the algorithm for any $z_k \in \mathbb{R}$ (it is straightforward to verify the error criterion as $x_k - z_k + \lambda \nabla f(x_k) = \sigma(y_k - z_k + \lambda \nabla f(x_k))$ in this case), and $\|z_{k+1}\| = \frac{1-\sigma+\lambda\mu\sigma}{1-\sigma+\lambda\mu}\|z_k\|$.

- The second term in the maximum is achieved by picking $f(x) = \frac{L}{2}x^2$ and $g(x) = i_{\{0\}}(x)$ (the indicator function of 0). In this case the solution is $x_* = z_* = 0$, and picking $x_k = \frac{1-\sigma}{1+(1-\sigma)L\lambda} z_k$, $y_k = 0$, and $z_{k+1} = \frac{\sigma+(1-\sigma)\lambda L}{1+(1-\sigma)\lambda L} z_k$ is a valid sequence for the algorithm for any $z_k \in \mathbb{R}$ (it is straightforward to verify the error criterion as $x_k - z_k + \lambda \nabla f(x_k) = \sigma(y_k - z_k + \lambda \nabla f(x_k))$ in this case), and $\|z_{k+1}\| = \frac{\sigma+(1-\sigma)\lambda L}{1+(1-\sigma)\lambda L}\|z_k\|$.

■

## 6.7 Conclusion

In this chapter, we proposed an inexact accelerated forward-backward method for solving composite convex minimization problems, along with some worst-case guarantees. The method supports inexact evaluations of the proximal subproblems, backtracking line-search on the smoothness parameter, and allows exploiting the possible strong convexity of one of the component in the objective function. The analysis relies on a now standard Lyapunov argument of the same type as that of Nesterov [1983], and the numerical behavior is illustrated on a factorization and a total variation problem. In addition, we provided a version of the A-HPE method [Monteiro and Svaiter, 2013] that supports possibly strongly convex objectives, and analyzed a partially inexact Douglas-Rachford splitting algorithm due to Eckstein and Yao [2018].

Let us note that the overall computational complexity of each numerical scheme presented in this chapter depends on the method used for obtaining an approximate solution for the proximal subproblem.

# Conclusion and Perspectives

Overall, this thesis took place in the global effort to better understand the behavior of first-order methods, through their worst-case analyses. For that purpose, we specifically relied on conic optimization, and on a principled approach to worst-case analysis, commonly referred to as *performance estimation problems* (PEPs). We extended the approach for being able to deal with adaptive first-order methods as well as approximate proximal operations. In addition, we analyzed a constrained nonlinear acceleration method.

We started with the study of an Anderson acceleration scheme with hard constraints on its extrapolation weights. When Anderson acceleration is applied to an affine linearly converging fixed point mapping (e.g. gradient steps on quadratic functions), its behavior is well understood. In order to provide guarantees for Anderson acceleration applied to nonlinear (non-affine) mappings, we performed a perturbation analysis. That is, seeing nonlinearity of the fixed point mapping as a perturbation of an affine one. In this nonlinear setting, we used similar polynomial arguments as in the linear case to obtain worst-case bounds. These guarantees followed from new upper-bounds on a constrained version of the Chebyshev optimization problem on polynomials. We believe those upper-bounds to be also of independent interest. However, we note that these robustness results are quite conservative due to the perturbation analysis. We believe that going beyond those results might require more advanced techniques not relying on perturbation arguments. Finding a way of simplifying performance estimation problems for Anderson acceleration schemes might constitute a possible solution.

Then, we focused on the worst-case analysis of adaptive first-order methods. In such methods, one typically combines first-order information using the current state of the algorithm, as opposed to fixed-step (or nonadaptive) methods that rely on predefined coefficients. Adaptive algorithms often exhibit better empirical performances than their nonadaptive counter parts, but lack of theoretical guarantees supporting these observations. We studied a class of adaptive methods based on Polyak step sizes and analyzed variants of gradient descent and accelerated gradient methods relying on these Polyak coefficients. To obtain worst-case guarantees we adapted the performance estimation approach from Drori and Teboulle [2014], Taylor et al. [2017c]. This allows using semidefinite programing for studying worst-case behavior of adaptive methods. We also illustrated this methodology by performing the worst-case analysis of a few adaptive algorithms numerically (namely gradient with exact line search, a nonlinear version of the celebrated conjugate gradient method, and a regularized Anderson acceleration algorithm).

Finally, we further extended the PEP framework to analyze first-order methods involving inexact proximal computations. Allowing inexactness in the proximal steps used inside optimization methods is a key feature for many algorithms. The "outer" algorithms we analyzed only require the approximate proximal steps to satisfy some inexactness criterion. In particular, they do not take into account how the approximated points were obtained. This allows producing generic analyses, as any strategy can be plugged-in to compute the proximal operations approximately, as long as it produces points satisfying

the inexactness requirements. However, this may lead to more conservative guarantees when studying the total cost of an inexact proximal method. Indeed, one may combine worst-case bounds of the inner method computing the proximal steps with worst-case guarantees of the "outer" algorithm.

We have seen that worst-case analysis is a powerful framework to study optimization algorithms. However it may be too conservative in some situations, e.g. for capturing the benefit brought by adaptive methods compared to nonadaptive ones. Other type of analyses might be better suited for such purposes. For instance, average-case analysis represents the behavior of an algorithm as a mean on a range of scenarios instead of the worst possible one. In Pedregosa and Scieur [2020], average-case analysis is performed for convex quadratic optimization by considering classes of functions represented by the distribution their Hessian's spectrum. It is not clear yet how to set tractable distributions on the class of smooth and strongly convex functions, but we believe that studying the average behaviors of optimization methods would reflect more finely their empirical performances.

We conclude by discussing possible new directions for the performance estimation methodology. We have seen that proofs derived from dual certificates of performance estimation problems consist in performing weighted sums of inequalities satisfied by elements of the functional class of interest, and showing that these sums have constant signs (by expressing them as sums of squares). This procedure is generic when working with fixed-step methods. In the case of adaptive methods, and in particular for gradient descent with Polyak step sizes, we observed that the corresponding PEP involved polynomials of degrees larger than 2 in function values and in kernelized variables (i.e. scalar product between iterates and/or gradients). In that case, one could look for worst-case guarantees through SOS decompositions of higher degrees, that is performing weighted sums of inequalities (that are polynomials in the kernelized variables) using polynomial weights and guaranteeing positivity using sum of squares certificates. We obtained underwhelming results when trying to apply these SOS tools to Polyak steps but we believe it might be worth looking deeper into.

# Bibliography


Hadi Abbaszadehpeivasti, Etienne de Klerk, and Moslem Zamani. The exact worst-case convergence rate of the gradient method with fixed step lengths for l-smooth functions. *arXiv preprint arXiv:2104.05468*, 2021.

Alexander Craig Aitken. On Bernoulli's numerical solution of algebraic equations. *Proceedings of the Royal Society of Edinburgh*, 46:289–305, 1927.

Amirhossein Ajalloeian, Andrea Simonetto, and Emiliano Dall'Anese. Inexact online proximal-gradient method for time-varying convex optimization. In *2020 American Control Conference (ACC)*, pages 2850–2857. IEEE, 2020.

Maicon M. Alves. Variants of the A-HPE and large-step A-HPE algorithms for strongly convex problems with applications to accelerated high-order tensor methods. *preprint arXiv:2102.02045 [V1]*, 2021.

Maicon M. Alves and Raul T. Marcavillaca. On inexact relative-error hybrid proximal extragradient, forward-backward and tseng's modified forward-backward methods with inertial effects. *Set-Valued and Variational Analysis*, pages 1–25, 2019.

Maicon M. Alves, Jonathan Eckstein, Marina Geremia, and Jefferson Melo. Relative-error inertial-relaxed inexact versions of Douglas-Rachford and ADMM splitting algorithms. *preprint arXiv:1904.10502*, 2019.

Donald G. Anderson. Iterative procedures for nonlinear integral equations. *Journal of the ACM (JACM)*, 12(4): 547–560, 1965.

Yossi Arjevani and Ohad Shamir. On the iteration complexity of oblivious first-order optimization algorithms. In *International Conference on Machine Learning*, pages 908–916, 2016.

Hilal Asi and John C. Duchi. Stochastic (approximate) proximal point methods: Convergence, optimality, and adaptivity. *SIAM Journal on Optimization*, 29(3):2257–2290, 2019.

Alexandre d' Aspremont, Damien Scieur, and Adrien Taylor. Acceleration methods. *preprint arXiv:2101.09545*, 2021.

Alfred Auslender. Numerical methods for nondifferentiable convex optimization. In *Nonlinear Analysis and Optimization*, pages 102–126. Springer, 1987.

Nikhil Bansal and Anupam Gupta. Potential-function proofs for gradient methods. *Theory of Computing*, 15(1): 1–32, 2019.

Jonathan Barzilai and Jonathan M. Borwein. Two-point step size gradient methods. *IMA journal of numerical analysis*, 8(1):141–148, 1988.

Nicola Bastianello, Amirhossein Ajalloeian, and Emiliano Dall'Anese. Distributed and inexact proximal gradient method for online convex optimization. *arXiv preprint arXiv:2001.00870*, 2020.

Heinz H. Bauschke and Patrick L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer, 2011.

Amir Beck. *First-order methods in optimization*. SIAM, 2017.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Yunier Bello-Cruz, Max L. N. Gonçalves, and Nathan Krislock. On inexact accelerated proximal gradient methods with relative error rules. *preprint arXiv:2005.03766*, 2020.

Dennis S. Bernstein. *Matrix mathematics*. Princeton university press, 2009.

Quentin Bertrand and Mathurin Massias. Anderson acceleration of coordinate descent. In *International Conference on Artificial Intelligence and Statistics*, pages 1288–1296. PMLR, 2021.

Raghu Bollapragada, Damien Scieur, and Alexandre d'Aspremont. Nonlinear acceleration of primal-dual algorithms. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 739–747, 2019.

Jérôme Bolte, Aris Daniilidis, and Adrian Lewis. The lojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17(4):1205–1223, 2007.

Jérôme Bolte, Trong Phong Nguyen, Juan Peypouquet, and Bruce W Suter. From error bounds to the complexity of first-order descent methods for convex functions. *Mathematical Programming*, 165(2):471–507, 2017.

Radu Ioan Boţ and Ernö Robert Csetnek. A hybrid proximal-extragradient algorithm with inertial effects. *Numerical Functional Analysis and Optimization*, 36(8):951–963, 2015.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Stephen Boyd, Lin Xiao, and Almir Mutapcic. Subgradient methods. *lecture notes of EE392o, Stanford University, Autumn Quarter*, 2004:2004–2005, 2003.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

Claude Brezinski. *Accélération de la convergence en analyse numérique*, volume 584. Springer, 2006.

Claude Brezinski and Michela Redivo-Zaglia. The genesis and early developments of aitken's process, shanks' transformation, the $\varepsilon$–algorithm, and related fixed point methods. *Numerical Algorithms*, 80(1):11–133, 2019.

Claude Brezinski, Stefano Cipolla, Michela Redivo-Zaglia, and Yousef Saad. Shanks and anderson-type acceleration techniques for systems of nonlinear equations. *arXiv preprint arXiv:2007.05716*, 2020.

Arne Brøndsted and Ralph T. Rockafellar. On the subdifferentiability of convex functions. *Proceedings of the American Mathematical Society*, 16(4):605–611, 1965.

Ronald E. Bruck Jr. An iterative solution of a variational inequality for certain monotone operators in Hilbert space. *Bulletin of the American Mathematical Society*, 81(5):890–892, 1975.

Regina S. Burachik, Alfredo N. Iusem, and Benar F. Svaiter. Enlargement of monotone operators with applications to variational inequalities. *Set-Valued Analysis*, 5(2):159–180, 1997.

Regina S. Burachik, Claudia A. Sagastizábal, and Benar F. Svaiter. $\varepsilon$-enlargements of maximal monotone operators: Theory and applications. In *Reformulation: nonsmooth, piecewise smooth, semismooth and smoothing methods*, pages 25–43. Springer, 1998.

Regina S. Burachik, Claudia A. Sagastizábal, and Benar F. Svaiter. Bundle methods for maximal monotone operators. In *Ill-posed variational problems and regularization techniques*, pages 49–64. Springer, 1999.

Regina S. Burachik, Juan Enrique Martínez-Legaz, Mahboubeh Rezaie, and Michel Théra. An additive subfamily of enlargements of a maximally monotone operator. *Set-Valued and Variational Analysis*, 23(4):643–665, 2015.

James V. Burke and Maijian Qian. A variable metric proximal point algorithm for monotone operators. *SIAM Journal on Control and Optimization*, 37(2):353–375, 1999.

Augustin Cauchy. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847.

Antonin Chambolle and Thomas Pock. An introduction to continuous optimization for imaging. *Acta Numerica*, 25:161–319, 2016.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Xiaojun Chen and Carl T. Kelley. Convergence of the EDIIS algorithm for nonlinear equations. *SIAM Journal on Scientific Computing*, 41(1):A365–A379, 2019.

Giovanni Chierchia, Emilie Chouzenoux, Patrick L. Combettes, and Jean-Christophe Pesquet. The proximity operator repository. user's guide, 2020. URL http://proximity-operator.net/download/guide.pdf.

Maxime Chupin, Mi-Song Dupuy, Guillaume Legendre, and Éric Séré. Convergence analysis of adaptive DIIS algorithms with application to electronic ground state calculations. *arXiv preprint arXiv:2002.12850*, 2020.

Patrick L Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

Roberto Cominetti. Coupling the proximal point algorithm with approximation methods. *Journal of Optimization Theory and Applications*, 95(3):581–600, 1997.

Rafael Correa and Claude Lemaréchal. Convergence of some algorithms for convex minimization. *Mathematical Programming*, 62(1-3):261–275, 1993.

Saman Cyrus, Bin Hu, Bryan Van Scoy, and Laurent Lessard. A robust accelerated optimization algorithm for strongly convex functions. In *2018 Annual American Control Conference (ACC)*, pages 1376–1381, 2018.

Yu-Hong Dai and Yaxiang Yuan. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM Journal on optimization*, 10(1):177–182, 1999.

Damek Davis and Wotao Yin. A three-operator splitting scheme and its optimization applications. *Set-Valued and Variational Analysis*, 25(4):829–858, 2017.

John E. Dennis and Jorge J. Moré. Quasi-newton methods, motivation and theory. *SIAM review*, 19(1):46–89, 1977.

John E. Dennis and Robert B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM, 1996.

Olivier Devolder. First-order methods with inexact oracle: the strongly convex case. *CORE Discussion Papers*, 2013.

Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1-2):37–75, 2014.

Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, 1997.

Rishabh Dixit, Amrit S. Bedi, Ruchi Tripathi, and Ketan Rajawat. Online learning with inexact proximal online gradient descent algorithms. *IEEE Transactions on Signal Processing*, 67(5):1338–1352, 2019.

Jim Douglas and Henry H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society*, 82:421–439, 1956.

Radu-Alexandru Dragomir, Adrien Taylor, Alexandre d'Aspremont, and Jérôme Bolte. Optimal complexity and certification of bregman first-order methods. *Mathematical Programming*, pages 1–43, 2021.

Yoel Drori. *Contributions to the Complexity Analysis of Optimization Algorithms*. PhD thesis, Tel-Aviv University, 2014.

Yoel Drori and Adrien Taylor. Efficient first-order methods for convex minimization: a constructive approach. *Mathematical Programming*, 184(1):183–220, 2020.

Yoel Drori and Marc Teboulle. Performance of first-order methods for smooth convex minimization: a novel approach. *Mathematical Programming*, 145(1-2):451–482, 2014.

Yoel Drori and Marc Teboulle. An optimal variant of Kelley's cutting-plane method. *Mathematical Program-*

*ming*, 160(1-2):321–351, 2016.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Jonathan Eckstein. *Splitting methods for monotone operators with applications to parallel optimization*. PhD thesis, Massachusetts Institute of Technology, 1989.

Jonathan Eckstein. Approximate iterations in bregman-function-based proximal algorithms. *Mathematical programming*, 83(1-3):113–123, 1998.

Jonathan Eckstein and Dimitri P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1-3):293–318, 1992.

Jonathan Eckstein and Paulo J. S. Silva. A practical relative error criterion for augmented lagrangians. *Mathematical Programming*, 141(1-2):319–348, 2013.

Jonathan Eckstein and Wang Yao. Augmented Lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results. *RUTCOR Research Reports*, 32(3), 2012.

Jonathan Eckstein and Wang Yao. Approximate ADMM algorithms derived from Lagrangian splitting. *Computational Optimization and Applications*, 68(2):363–405, 2017.

Jonathan Eckstein and Wang Yao. Relative-error approximate versions of Douglas–Rachford splitting and special cases of the ADMM. *Mathematical Programming*, 170(2):417–444, 2018.

Haw-Ren Fang and Yousef Saad. Two classes of multisecant methods for nonlinear acceleration. *Numerical Linear Algebra with Applications*, 16(3):197–221, 2009.

Olivier Fercoq and Zheng Qu. Restarting accelerated gradient methods with a rough strong convexity estimate. *arXiv preprint arXiv:1609.07358*, 2016.

Donald A. Flanders and George Shortley. Numerical determination of fundamental modes. *Journal of Applied Physics*, 21(12):1326–1332, 1950.

Roger Fletcher. Practical methods of optimization, 1987.

Roger Fletcher. On the Barzilai-Borwein method. In *Optimization and control with applications*, pages 235–256. Springer, 2005.

Roger Fletcher and Colin M. Reeves. Function minimization by conjugate gradients. *The computer journal*, 7 (2):149–154, 1964.

Michel Fortin and Roland Glowinski. On decomposition-coordination methods using an Augmented Lagrangian. In M. Fortin and R. Glowinski, editors, *Augmented Lagrangian Methods: Applications to the Solution of Boundary-Value Problems*. North-Holland:Amsterdam, 1983.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

Anqi Fu, Junzi Zhang, and Stephen Boyd. Anderson accelerated douglas–rachford splitting. *SIAM Journal on Scientific Computing*, 42(6):A3560–A3583, 2020.

Marc Fuentes, Jérôme Malick, and Claude Lemaréchal. Descentwise inexact proximal algorithms for smooth optimization. *Computational Optimization and Applications*, 53(3):755–769, 2012.

Daniel Gabay. Applications of the method of multipliers to variational inequalities. In M. Fortin and R. Glowinski, editors, *Augmented Lagrangian Methods: Applications to the Solution of Boundary-Value Problems*. North-Holland:Amsterdam, 1983.

Pontus Giselsson and Stephen Boyd. Diagonal scaling in douglas-rachford splitting and admm. In *53rd IEEE Conference on Decision and Control*, pages 5033–5039. IEEE, 2014.

Pontus Giselsson and Stephen Boyd. Linear convergence and metric selection for Douglas–Rachford splitting and ADMM. *IEEE Transactions on Automatic Control*, 62(2):532–544, 2016.

Gene H. Golub and Charles F. Van Loan. Matrix computation. *North Oxford Academic*, 1990.

Gene H. Golub and Richard S. Varga. Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order richardson iterative methods. *Numerische Mathematik*, 3(1):157–168, 1961.

Paul R. Gorman and Terrence J. Sejnowski. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1:75, 1988.

Robert M. Gower, Aaron Defazio, and Michael Rabbat. Stochastic polyak stepsize with a moving target. *arXiv preprint arXiv:2106.11851*, 2021.

Guoyong Gu and Junfeng Yang. Optimal nonergodic sublinear convergence rate of proximal point algorithm for maximal monotone inclusion problems. *preprint arXiv:1904.05495*, 2019a.

Guoyong Gu and Junfeng Yang. On the optimal ergodic sublinear convergence rate of the relaxed proximal point algorithm for variational inequalities. *preprint arXiv:1905.06030*, 2019b.

Osman Güler. New proximal point algorithms for convex minimization. *SIAM Journal on Optimization*, 2(4): 649–664, 1992.

William W. Hager and Hongchao Zhang. A survey of nonlinear conjugate gradient methods. *Pacific journal of Optimization*, 2(1):35–58, 2006.

Elad Hazan and Sham Kakade. Revisiting the Polyak step size. *arXiv preprint arXiv:1905.00313*, 2019.

Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49(6):409–436, 1952.

Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Convex analysis and minimization algorithms I: Fundamentals*, volume 305. Springer science & business media, 2013.

Bin Hu and Laurent Lessard. Dissipativity theory for nesterov's accelerated method. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1549–1557. JMLR. org, 2017.

Alfredo N. Iusem. Augmented Lagrangian methods and proximal point methods for convex optimization. *Investigación Operativa*, 8(11-49):7, 1999.

Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *ICML (1)*, pages 427–435, 2013.

Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, and Francis Bach. Proximal methods for sparse hierarchical dictionary learning. In *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML)*, pages 487–494, 2010.

Kaifeng Jiang, Defeng Sun, and Kim-Chuan Toh. An inexact accelerated proximal gradient method for large scale linearly constrained convex sdp. *SIAM Journal on Optimization*, 22(3):1042–1064, 2012.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016.

Donghwan Kim. Accelerated proximal point method for maximally monotone operators. *Mathematical Programming*, pages 1–31, 2021.

Donghwan Kim and Jeffrey A. Fessler. Optimized first-order methods for smooth convex minimization. *Mathematical Programming*, 159(1-2):81–107, 2016.

Donghwan Kim and Jeffrey A Fessler. Another look at the fast iterative shrinkage/thresholding algorithm (fista). *SIAM Journal on Optimization*, 28(1):223–250, 2018.

Donghwan Kim and Jeffrey A Fessler. Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions. *Journal of Optimization Theory and Applications*, 188(1):192–219, 2021.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Etienne de Klerk, François Glineur, and Adrien Taylor. On the worst-case complexity of the gradient method with exact line search for smooth strongly convex functions. *Optimization Letters*, 11(7):1185–1199, 2017.

Etienne de Klerk, Francois Glineur, and Adrien Taylor. Worst-case convergence analysis of inexact gradient and newton methods through semidefinite programming performance estimation. *SIAM Journal on Optimization*, 30(3):2053–2082, 2020.

Jean-Bernard Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

Jongmin Lee, Chanwoo Park, and Ernest K. Ryu. A geometric structure of acceleration and its role in making gradients small fast. *arXiv preprint arXiv:2106.10439*, 2021.

Bernard Lemaire. About the convergence of the proximal method. In *Advances in Optimization*, pages 39–51. Springer, 1992.

Claude Lemaréchal. Cauchy and the gradient method. *Doc Math Extra*, 251(254):10, 2012.

Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.

Felix Lieder. On the convergence rate of the halpern-iteration. *Optimization Letters*, pages 1–14, 2020.

Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, pages 3384–3392, 2015.

Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. Catalyst acceleration for first-order convex optimization: from theory to practice. *Journal of Machine Learning Research*, 18(212):1–54, 2018.

Qihang Lin and Lin Xiao. An adaptive accelerated proximal gradient method and its homotopy continuation for sparse optimization. In *ICML*, pages 73–81, 2014.

Pierre-Louis Lions and Bertrand Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.

Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

Johan Löfberg. YALMIP : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, 2004.

Nicolas Loizou, Sharan Vaswani, Issam Hadj Laradji, and Simon Lacoste-Julien. Stochastic polyak step-size for sgd: An adaptive learning rate for fast convergence. In *International Conference on Artificial Intelligence and Statistics*, pages 1306–1314. PMLR, 2021.

Aleksandr M. Lyapunov and Anthony T. Fuller. *General Problem of the Stability Of Motion*. Control Theory and Applications Series. Taylor & Francis, 1992. Original text in Russian, 1892.

Victor Magron, Mohab Safey El Din, and Markus Schweighofer. Algorithms for weighted sum of squares decomposition of non-negative univariate polynomials. *Journal of Symbolic Computation*, 93:200–220, 2019.

Vien Mai and Mikael Johansson. Anderson acceleration of proximal gradient methods. In *International Conference on Machine Learning*, pages 6620–6629. PMLR, 2020.

Julien Mairal, Rodolphe Jenatton, Guillaume Obozinski, and Francis Bach. Convex and network flow optimization for structured sparsity. *Journal of Machine Learning Research*, 12(Sep):2681–2720, 2011.

Yura Malitsky and Konstantin Mishchenko. Adaptive gradient descent without descent. *arXiv preprint arXiv:1910.09529*, 2019.

Bernard Martinet. Régularisation d'inéquations variationnelles par approximations successives. *Revue Française d'Informatique et de Recherche Opérationnelle*, 4:154–158, 1970.

Bernard Martinet. Détermination approchée d'un point fixe d'une application pseudo-contractante. cas de l'application prox. *Comptes rendus hebdomadaires des séances de l'Académie des sciences de Paris*, 274: 163–165, 1972.

John C. Mason and David C. Handscomb. *Chebyshev polynomials*. CRC press, 2002.

Mathurin Massias, Alexandre Gramfort, and Joseph Salmon. Celer: a fast solver for the lasso with dual extrapolation. In *International Conference on Machine Learning*, pages 3315–3324, 2018.

Alexandre Megretski and Anders Rantzer. System analysis via integral quadratic constraints. *IEEE Transactions on Automatic Control*, 42(6):819–830, 1997.

Reinier D. Millán and Mariela P. Machado. Inexact proximal $epsilon$-subgradient methods for composite convex optimization problems. *Journal of Global Optimization*, 75(4):1029–1060, 2019.

Renato D.C. Monteiro and Benar F. Svaiter. On the complexity of the hybrid proximal extragradient method for the iterates and the ergodic mean. *SIAM Journal on Optimization*, 20(6):2755–2787, 2010.

Renato D.C. Monteiro and Benar F. Svaiter. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM Journal on Optimization*, 23(2):1092–1125, 2013.

Jean-Jacques Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *Comptes rendus hebdomadaires des séances de l'Académie des sciences de Paris*, 255:2897–2899, 1962.

Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.

APS Mosek. The MOSEK optimization software. *Online at http://www.mosek.com*, 54, 2010.

Angelia Nedic and Dimitri P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM Journal on Optimization*, 12(1):109–138, 2001.

Arkadi S. Nemirovski. Prox-method with rate of convergence $o(1/t)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

Arkadi S. Nemirovskii and Yuri Nesterov. Optimal methods of smooth convex minimization. *USSR Computational Mathematics and Mathematical Physics*, 25(2):21–30, 1985.

Arkadi S. Nemirovskiy and Boris T. Polyak. Iterative methods for solving linear ill-posed problems under precise information. *ENG. CYBER.*, (4):50–56, 1984.

Arkadi S. Nemirovsky. Information-based complexity of linear operator equations. *Journal of Complexity*, 8(2): 153–175, 1992.

Arkadi S. Nemirovsky. Information-based complexity of convex programming. *Lecture Notes*, 1995. URL https://www2.isye.gatech.edu/~nemirovs/Lec_EMCO.pdf.

Yurii Nesterov. A method of solving a convex programming problem with convergence rate O($1/k^2$). *Soviet Mathematics Doklady*, 27:372–376, 1983.

Yurii Nesterov. *Introductory Lectures on Convex Optimization : a Basic Course*. Applied optimization. Kluwer Academic Publishing, 2004.

Yurii Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1): 125–161, 2013.

Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

Yurii Nesterov. Implementable tensor methods in unconstrained convex optimization. *Mathematical Programming*, pages 1–27, 2019.

Yurii Nesterov. Inexact high-order proximal-point methods with auxiliary search procedure. Technical report, CORE discussion paper, 2020a.

Yurii Nesterov. Inexact accelerated high-order proximal-point methods. Technical report, CORE discussion paper, 2020b.

Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

Brendan O'Donoghue and Emmanuel Candes. Adaptive restart for accelerated gradient schemes. *Foundations of computational mathematics*, 15(3):715–732, 2015.

Wenqing Ouyang, Jiong Tao, Andre Milzarek, and Bailin Deng. Nonmonotone globalization for anderson acceleration using adaptive regularization. *arXiv preprint arXiv:2006.02559*, 2020.

Daniel Palomar and Yonina Eldar. *Convex optimization in signal processing and communications*. Cambridge university press, 2010.

Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

Pablo A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, 2000.

Gregory B. Passty. Ergodic convergence to a zero of the sum of monotone operators in Hilbert space. *Journal of Mathematical Analysis and Applications*, 72(2):383–390, 1979.

Fabian Pedregosa and Damien Scieur. Acceleration through spectral density estimation. In *International Conference on Machine Learning*, pages 7553–7562. PMLR, 2020.

Sara Pollock and Leo Rebholz. Anderson acceleration for contractive and noncontractive operators. *arXiv preprint arXiv:1909.04638*, 2019.

Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Boris T. Polyak. The conjugate gradient method in extremal problems. *USSR Computational Mathematics and Mathematical Physics*, 9(4):94–112, 1969.

Boris T. Polyak. *Introduction to optimization*. Optimization Software, New York, 1987.

Clarice Poon and Jingwei Liang. Trajectory of alternating direction method of multipliers and adaptive acceleration. In *Advances in Neural Information Processing Systems*, pages 7355–7363, 2019.

Péter Pulay. Convergence acceleration of iterative sequences. the case of scf iteration. *Chemical Physics Letters*, 73(2):393–398, 1980.

Mihai Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42(3):969–984, 1993.

James Renegar and Benjamin Grimmer. A simple nearly optimal restart scheme for speeding up first-order methods. *Foundations of Computational Mathematics*, pages 1–46, 2021.

Ralph T. Rockafellar. A dual approach to solving nonlinear programming problems by unconstrained optimization. *Mathematical programming*, 5(1):354–373, 1973.

Ralph T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of operations research*, 1(2):97–116, 1976a.

Ralph T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976b.

Ralph T. Rockafellar. *Convex Analysis*. Princeton University Press, 1996.

William Rodi and Randall L. Mackie. Nonlinear conjugate gradients algorithm for 2-d magnetotelluric inversion. *Geophysics*, 66(1):174–187, 2001.

Thorsten Rohwedder and Reinhold Schneider. An analysis for the DIIS acceleration method used in quantum chemistry calculations. *Journal of mathematical chemistry*, 49(9):1889, 2011.

Vincent Roulet and Alexandre d'Aspremont. Sharpness, restart, and acceleration. *SIAM Journal on Optimization*, 30(1):262–289, 2020.

Leonid I. Rudin and Stanley Osher. Total variation based image restoration with free local constraints. In *Proceedings of 1st International Conference on Image Processing*, volume 1, pages 31–35. IEEE, 1994.

Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

Ernest K. Ryu and Stephen Boyd. Primer on monotone operator methods. *Applied and Computational Mathematics*, 15(1):3–43, 2016.

Ernest K. Ryu and Bang Công Vũ. Finding the forward-Douglas–Rachford-forward method. *Journal of Optimization Theory and Applications*, pages 1–19, 2019.

Ernest K. Ryu, Robert Hannah, and Wotao Yin. Scaled relative graph: Nonexpansive operators via 2d euclidean geometry. *preprint arXiv:1902.09788*, 2019.

Ernest K. Ryu, Adrien Taylor, Carolina Bergeling, and Pontus Giselsson. Operator splitting performance estimation: Tight contraction factors and optimal parameter selection. *SIAM Journal on Optimization*, 30(3): 2251–2271, 2020.

Youcef Saad and Martin H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.

Saverio Salzo and Silvia Villa. Inexact and accelerated proximal point algorithms. *Journal of Convex Analysis*, 19(4):1167–1192, 2012.

Katya Scheinberg, Donald Goldfarb, and Xi Bai. Fast first-order methods for composite convex optimization with backtracking. *Foundations of Computational Mathematics*, 14(3):389–417, 2014.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Advances in neural information processing systems (NIPS)*, pages 1458–1466, 2011.

Damien Scieur, Alexandre d'Aspremont, and Francis Bach. *Regularized Nonlinear Acceleration*. NIPS, 2016.

Damien Scieur, Alexandre d'Aspremont, and Francis Bach. Nonlinear acceleration of stochastic algorithms. *arXiv preprint arXiv:1706.07270*, 2017.

Damien Scieur, Edouard Oyallon, Alexandre d'Aspremont, and Francis Bach. Online regularized nonlinear acceleration. *arXiv preprint arXiv:1805.09639*, 2018.

Damien Scieur, Alexandre d'Aspremont, and Francis Bach. Regularized nonlinear acceleration. *Mathematical Programming*, 179(1):47–83, 2020.

Daniel Shanks. Non-linear transformations of divergent and slowly convergent sequences. *Journal of Mathematics and Physics*, 34(1):1–42, 1955.

Jonathan R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain, 1994.

Avram Sidi, William F. Ford, and David A. Smith. Acceleration of convergence of vector sequences. *SIAM Journal on Numerical Analysis*, 23(1):178–196, 1986.

Andrea Simonetto and Hadi Jamali-Rad. Primal recovery from consensus-based dual decomposition for distributed convex optimization. *Journal of Optimization Theory and Applications*, 168(1):172–197, 2016.

David A. Smith, William F. Ford, and Avram Sidi. Extrapolation methods for vector sequences. *SIAM review*, 29(2):199–233, 1987.

Michael V. Solodov and Benar F. Svaiter. A comparison of rates of convergence of two inexact proximal point algorithms. In *Nonlinear optimization and related topics*, pages 415–427. Springer, 2000a.

Mikhail V. Solodov and Benar F. Svaiter. A hybrid approximate extragradient–proximal point algorithm using the enlargement of a maximal monotone operator. *Set-Valued Analysis*, 7(4):323–345, 1999a.

Mikhail V. Solodov and Benar F. Svaiter. A hybrid projection-proximal point algorithm. *Journal of Convex Analysis*, 6(1):59–70, 1999b.

Mikhail V. Solodov and Benar F. Svaiter. Error bounds for proximal point subproblems and associated inexact proximal point algorithms. *Mathematical programming*, 88(2):371–389, 2000b.

Mikhail V. Solodov and Benar F. Svaiter. An inexact hybrid generalized proximal point algorithm and some new

results on the theory of Bregman functions. *Mathematics of Operations Research*, 25(2):214–230, 2000c.

Mikhail V. Solodov and Benar F. Svaiter. A unified framework for some inexact proximal point algorithms. *Numerical functional analysis and optimization*, 22(7-8):1013–1035, 2001.

Jos F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

Benar F. Svaiter. A weakly convergent fully inexact Douglas-Rachford method with relative error tolerance. *preprint arXiv:1809.02312*, 2018.

Adrien Taylor. *Convex interpolation and performance estimation of first-order methods for convex optimization*. PhD thesis, Université catholique de Louvain, 2017.

Adrien Taylor and Francis Bach. Stochastic first-order methods: non-asymptotic and computer-aided analyses via potential functions. In *Proceedings of the Thirty-Second Conference on Learning Theory (COLT)*, volume 99, pages 2934–2992. PMLR, 2019.

Adrien Taylor and Yoel Drori. An optimal gradient method for smooth (possibly strongly) convex minimization. *arXiv preprint arXiv:2101.09741*, 2021.

Adrien Taylor, Julien M. Hendrickx, and François Glineur. Exact worst-case performance of first-order methods for composite convex optimization. *SIAM Journal on Optimization*, 27(3):1283–1313, 2017a.

Adrien Taylor, Julien M. Hendrickx, and François Glineur. Performance Estimation Toolbox (PESTO): automated worst-case analysis of first-order optimization methods. In *IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1278–1283, 2017b.

Adrien Taylor, Julien M. Hendrickx, and François Glineur. Smooth strongly convex interpolation and exact worst-case performance of first-order methods. *Mathematical Programming*, 161(1-2):307–345, 2017c.

Adrien Taylor, Julien M. Hendrickx, and François Glineur. Exact worst-case convergence rates of the proximal gradient method for composite convex minimization. *Journal of Optimization Theory and Applications*, 178 (2):455–476, 2018a.

Adrien Taylor, Bryan Van Scoy, and Laurent Lessard. Lyapunov functions for first-order methods: Tight automated convergence guarantees. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 4897–4906, 2018b.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Kim-Chuan Toh, Michael J. Todd, and Reha H. Tütüncü. On the implementation and usage of SDPT3–a Matlab software package for semidefinite-quadratic-linear programming, version 4.0. In *Handbook on semidefinite, conic and polynomial optimization*, pages 715–754. Springer, 2012.

Onur Toker and Hitay Ozbay. On the np-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback. In *1995 Annual American Control Conference (ACC)*, volume 4, pages 2525–2526, 1995.

Alex Toth and Carl T. Kelley. Convergence analysis for anderson acceleration. *SIAM Journal on Numerical Analysis*, 53(2):805–819, 2015.

Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal on Optimization*, 2008.

Bryan Van Scoy, Randy A. Freeman, and Kevin M. Lynch. The fastest known globally convergent first-order method for minimizing strongly convex functions. *IEEE Control Systems Letters*, 2(1):49–54, 2018.

Richard S. Varga. *Iterative analysis*. Springer, 1962.

Silvia Villa, Saverio Salzo, Luca Baldassarre, and Alessandro Verri. Accelerated and inexact forward-backward algorithms. *SIAM Journal on Optimization*, 23(3):1607–1633, 2013.

Stefan Volkwein. Nonlinear conjugate gradient methods for the optimal control of laser surface hardening.

*Optimization Methods and Software*, 19(2):179–199, 2004.

Yilun Wang, Junfeng Yang, Wotao Yin, and Yin Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008.

Ashia Wilson. *Lyapunov arguments in optimization*. PhD thesis, UC Berkeley, 2018.

Ashia Wilson, Ben Recht, and Michael I. Jordan. A Lyapunov analysis of accelerated methods in optimization. *Journal of Machine Learning Research*, 22(113):1–34, 2021.

Junzi Zhang, Brendan O'Donoghue, and Stephen Boyd. Globally convergent type-i anderson acceleration for nonsmooth fixed-point iterations. *SIAM Journal on Optimization*, 30(4):3170–3197, 2020.

Jing Zhao, Edwin Vollebregt, and Cornelis W. Oosterlee. A fast nonlinear conjugate gradient based method for 3d concentrated frictional contact problems. *Journal of Computational Physics*, 288:86–100, 2015.

Chunxiang Zong, Yuchao Tang, and Yeol Cho. Convergence analysis of an inexact three-operator splitting algorithm. *Symmetry*, 10(11):563, 2018.

## RÉSUMÉ

De nombreuses applications modernes reposent sur la résolution de problèmes d'optimisations (par exemple, en biologie numérique, en mécanique, en finance), faisant des méthodes d'optimisation des outils essentiels dans de nombreux domaines scientifiques. Apporter des garanties sur le comportement de ces méthodes constitue donc un axe de recherche important.
Une façon classique d'analyser un algorithme d'optimisation consiste à étudier son comportement dans le pire cas. C'est à dire, donner des garanties sur son comportement (par exemple sa vitesse de convergence) qui soient indépendantes de la fonction en entrée de l'algorithme et vraies pour toutes les fonctions dans une classe donnée. Cette thèse se concentre sur l'analyse en pire cas de quelques méthodes du premier ordre réputées pour leur efficacité.
Nous commençons par étudier les méthodes d'accélération d'Anderson, pour lesquelles nous donnons de nouvelles bornes de pire cas qui permettent de garantir précisément et explicitement quand l'accélération a lieu. Pour obtenir ces garanties, nous fournissons des majorations sur une variation du problème d'optimisation polynomiale de Tchebychev, dont nous pensons qu'elles constituent un résultat indépendant.
Ensuite, nous prolongeons l'étude des Problèmes d'Estimation de Performances (PEP), développés à l'origine pour analyser les algorithmes d'optimisation à pas fixes, à l'analyse des méthodes adaptatives. En particulier, nous illustrons ces développements à travers l'étude des comportements en pire cas de la descente de gradient avec pas de Polyak, qui utilise la norme des gradients et les valeurs prises par la fonction objectif, ainsi que d'une nouvelle version accélérée. Nous détaillons aussi cette approche sur d'autres algorithmes adaptatifs standards.
Enfin, la dernière contribution de cette thèse est de développer plus avant la méthodologie PEP pour l'analyse des méthodes du premier ordre se basant sur des opérations proximales inexactes. En utilisant cette approche, nous définissons des algorithmes dont les garanties en pire cas ont été optimisées et nous fournissons des analyses de pire cas pour quelques méthodes présentes dans la littérature.

## MOTS CLÉS

optimisation convexe, méthodes adaptatives, problèmes d'estimation de performances, analyse de pire cas, taux de convergence

## ABSTRACT

Many modern applications rely on solving optimization problems (e.g., computational biology, mechanics, finance), establishing optimization methods as crucial tools in many scientific fields. Providing guarantees on the (hopefully good) behaviors of these methods is therefore of significant interest.
A standard way of analyzing optimization algorithms consists in worst-case reasoning. That is, providing guarantees on the behavior of an algorithm (e.g. its convergence speed), that are independent of the function on which the algorithm is applied and true for every function in a particular class. This thesis aims at providing worst-case analyses of a few efficient first-order optimization methods.
We start by the study of Anderson acceleration methods, for which we provide new explicit worst-case bounds guaranteeing precisely when acceleration occurs. We obtained these guarantees by providing upper bounds on a variation of the classical Chebyshev optimization problem on polynomials, that we believe of independent interest.
Then, we extend the Performance Estimation Problem (PEP) framework, that was originally designed for principled analyses of fixed-step algorithms, to study first-order methods with adaptive parameters. This is illustrated in particular through the worst-case analyses of the canonical gradient method with Polyak step sizes that use gradient norms and function values information, and of an accelerated version of it. The approach is also presented on other standard adaptive algorithms.
Finally, the last contribution of this thesis is to further develop the PEP methodology for analyzing first-order methods relying on inexact proximal computations. Using this framework, we produce algorithms with optimized worst-case guarantees and provide (numerical and analytical) worst-case bounds for some standard algorithms in the literature.

## KEYWORDS

convex optimization, adaptive methods, performance estimation problems, worst-case analysis, convergence rates