Università degli Studi di Ferrara

Dottorato di Ricerca in Matematica

Ciclo XXIX, Coordinatore Prof. Massimiliano Mella

Variable metric line–search based methods for nonconvex optimization

Settore Scientifico Disciplinare MAT/08

Dottorando	Tutore	Co-Tutore
Dr. Simone Rebegoldi	Dr. Marco Prato	Dr. Silvia Bonettini

Anni 2014/2016

Contents

Acknowledgements

This thesis is the result of the doctoral research I have undertaken during the last three years. It has been an amazing journey in the world of optimization and image processing that has enriched me on many levels. For that reason, I feel the need to express my gratitude to those who made this journey possible, in a way or another.

First of all, I am greatly indebted to my advisor Marco Prato for his valuable guidance and support throughout my doctoral studies. Marco introduced me to the world of inverse problems in the best possible way and taught me how to tackle a problem in research. He kept my motivation high with his clever point of view on each and every aspect of our work. For all of that, I am very grateful to him. Likewise, I would like to thank my co-advisor Silvia Bonettini for the meaningful discussions on the algorithmic and theoretical issues related to our work, which gave me a clear view on what we were doing and pushed me to do my best. I thank Luca Zanni for sharing with me his wide knowledge of optimization and numerical analysis, as well as crucially contributing to the research included in Chapter 4. I also thank Valeria Ruggiero, whose dedition to research is admirable and has greatly inspired me during my PhD, and Gaetano Zanghirati, who I admire for his generosity in sharing his knowledge and skills with his students and colleagues. A special thank goes to Laure Blanc-Féraud for her warm ospitality during my three-month period at the Laboratoire I3S, Université Nice Sophia Antipolis, her thrilling way of doing research and for suggesting me to work together on DIC microscopy.

None of the work included in this thesis would have been possible without my family. My mother Mariuccia deserves all my gratitude for the enormous amount of love, patience and continuous support she has been giving me over the years. To my father Ferdinando, I hope that you are proud of me and see my defense from where you are now. My brother Francesco and my sister Chiara have been my role models since I was a child, and will always be. You are the best brother/sister in the world! Also, I basically consider my nephew Leonardo as my little brother and my brother-in-law Giampietro as one of my best friends. To all of you, thank you for shaping me into the person I am today.

I want to thank my fellows at the PhD office in Modena, i.e. in order of appearance Anastasia, Federica, Tatiana, Alessandro, Roberto, Gennaro, Vanna, Chiara, Elena, Matteo, Lorenzo, Carla, Serena for making this journey in the research world funnier and less stressful. Many thanks also to Lola, who has been a great research teammate during (and after) my stay in Sophia Antipolis, and a big HUIII to all the fellows that I have met at the I3S Lab. A very special thank goes to the Fantastic Week Wi-Fi Gratis Whatsapp group, starring Riccardo (Ciba), Denise (Deee andiamo a casa), Elena (Yelena), Fabio (Fargnan C.O.), Ivan (Drago), Lorenzo (Senior), Paolo (Mambro), Francesco (Junior), Barbara, Fabio (Cava), Cinzia, Margherita (Maggie), Federico (Fede) for the countless lunches and dinners, the laughs, the jokes and all the amazing moments we had together during the last three years.

The last person I am going to thank has taught me the most important thing of all, the one that you can not learn from books, prove by contradiction or simulate on a computer, that is love. Thank you Elja for making me fall in love with you.

Introduction

Nonlinear optimization has increasingly become relevant in the resolution of problems arising from several domains of applied science. Indeed many phenomena occurring in applications involve the minimization of a nonlinear objective function which typically depends on a large number of unknown variables. Most of the times, such minimization problem is not solvable exactly, which makes necessary to compute a numerical approximation of the solution via an iterative algorithm. The main challenge of nonlinear optimization consists then in devising algorithms which are able to provide accurate estimations of the optimal solution in a reasonable amount of time.

First order methods have gained much popularity in the optimization framework. When the objective function is differentiable, these algorithms are denominated *gradient methods* and involve the iterative evaluation of only the function and its gradient until convergence to the solution is achieved. Projection of the iterates onto a convex set may be required whenever constraints are imposed on the unknown variables, thus leading to the rise of *gradient projection methods*. The natural extension of these algorithms to nondifferentiable problems is represented by the so-called *proximal–gradient methods*, which are applicable when the objective function is given by the sum of a differentiable term and a convex term.

What makes first order methods so attractive is their simplicity of use and low computational cost per iteration. The downside is that they often exhibit a slow rate of convergence to the solution of the optimization problem. Therefore, acceleration strategies have been devised in the literature in order to turn first order methods into competitive and efficient tools. Several of them are related to the adaptive choice of the parameters involved in the definition of the algorithm, such as the *steplength parameter* or the *scaling matrix* defining the descent direction. In particular, steplength selection rules are usually based on the information available at the previous iterations, whereas the scaling matrix is chosen according to the features and shape of the objective function. Together, these parameters define the *variable metric* with respect to which the iterate is computed.

Another major difficulty in devising effective algorithms is nonconvexity. When convexity is lost, the objective function presents multiple local minima and possibly saddle points. Furthermore, in this case only global convergence (in the sense of subsequences) to stationary points is usually guaranteed for first order methods. These theoretical remarks lead to an overall uncertainty in numerical experience on whether the algorithm of interest is actually converging and, in this case, if it converges to a sensible estimate of the solution.

The aim of this thesis is to propose efficient first order methods tailored for a wide class of nonconvex nondifferentiable optimization problems, in which the objective function is given by the sum of a differentiable, possibly nonconvex function and a convex, possibly nondifferentiable term. Our approach is twofold: on one hand, we accelerate the proposed methods by making use of suitable adaptive strategies to choose the involved parameters; on the other hand, we ensure convergence by imposing a sufficient decrease condition on the objective function.

Our first contribution is the development of a novel proximal–gradient method denominated Variable Metric Inexact Line–search based Algorithm (VMILA). The proposed approach is innovative from several points of view. First of all, VMILA allows to adopt a variable metric in the computation of the proximal point with a relative freedom of choice. Indeed the only assumption that we make is that the parameters involved belong to bounded sets. This is unusual with respect to the state-of-the-art proximal–gradient methods, where the parameters are usually chosen by means of a fixed rule or tightly related to the Lipschitz constant of the problem. Second, we introduce an inexactness criterion for computing the proximal point which can be practically implemented in some cases of interest. This aspect assumes a relevant importance whenever the proximal operator is not available in a closed form, which is often the case. Third, the VMILA iterates are computed by performing a line–search along the feasible direction and according to a specific Armijo-like condition. This last one can be indeed considered as an extension of the classical Armijo rule proposed in the context of differentiable optimization.

The VMILA method has been originally proposed in [32], in which the convergence and the numerical experience are shown only in the convex case. In this thesis, we propose a suitable modification of this method, denominate VMILAn, for which the convergence analysis can be extended to the nonconvex case. As a first result, we prove that each limit point of the VMILAn sequence is stationary for the objective function, provided that the gradient of the differentiable part is Lipschitz continuous. In the second place, we show that the sequence converges to a stationary point by assuming that the objective function satisfies a very general analytical property, the so-called *Kurdyka–Łojasiewicz inequality* (KL). This condition is satisfied by a large variety of functions and requires a certain regular behaviour of the function in a neighbourhood of its critical points. The proof of this result is essentially obtained by combining the properties of the Armijo line–search with the KL property. Finally, we also prove some convergence rate results for the VMILAn sequence, according to the degree of regularity specified by the KL property.

The numerical efficiency of the proposed approach is then shown on a collection of nonconvex problems arising in image processing. In particular, we observe how VMILAn is able to provide accurate reconstruction of the unknown image in a lower computational time, in some cases of several order of magnitudes, with respect to other accelerated proximal–gradient methods known in the literature. In the related comments, we conjecture that this accelerated rate of convergence may be due more to the variable choice of the parameters involved than the sufficient decrease condition imposed by the Armijo-like condition.

The effectiveness of VMILAn is further demonstrated by treating in details the problem of phase estimation arising in differential-interference-contrast (DIC) microscopy. Such problem consists in recovering information on the phase shifts of the light induced by the specimen from a set of images acquired with a DIC microscope. The resulting optimization problem is highly nonconvex and, thus, can be perfectly addressed by means of VMILAn. Unlike previous works on DIC, we decide to adopt an edge-preserving approach to this problem and propose to regularize it by means of the Hypersurface (HS) potential, which is a possibly smoothed version of the Total Variation (TV) functional. The DIC problem is then tackled in two ways: in the case of HS regularization, a gradient method equipped with an Armijo line–search and a non standard selection rule for the steplength is adopted; in the case of TV regularization, a non-scaled version of VMILAn is considered. Numerical simulations with simulated datasets show that the two proposed line–search based techniques are able to provide accurate reconstructions of the phase in a lower computational time than several nonlinear conjugate gradient methods, including the state-of-the-art method for DIC imaging.

In our second contribution we treat a special instance of the previously considered optimization problem, where the convex term is assumed to be a finite sum of the indicator functions of closed, convex sets. In other words, we consider a problem of constrained differentiable optimization in which the constraints have a separable structure. The most suited method to deal with this problem is undoubtedly the *nonlinear Gauss-Seidel* (GS) or *block coordinate descent method*, where the minimization of the objective function is cyclically alternated on each block of variables of the problem. In this thesis, we propose an inexact version of the GS scheme, denominated Cyclic Block Generalized Gradient Projection (CBGGP) method, in which the partial minimization over each block of variables is performed inexactly by means of a fixed number of gradient projection steps. The novelty of the proposed approach consists in the introduction of non Euclidean metrics in the computation of the gradient projection. The general result that we provide for CBGGP is the stationarity of the limit points of the sequence, without any convexity assumption on the objective function. Furthermore, we extensively apply CBGGP in large-scale image blind deconvolution, when the data are corrupted by either Gaussian or Poisson noise. Finally, we deepen the numerical study of CBGGP on a series of realistic simulations in blind deconvolution problems for ground-based telescopes equipped with Fizeau interferometers, in which successes and failures in the reconstruction of stellar fields are shown.

The thesis is organized as follows.

In Chapter 1, we introduce the reader to the main concepts of differentiable optimization and provide an insightful overview of gradient methods. In particular, we discuss the choice of both the steplength and the scaling matrix, in both the unconstrained and constrained setting, and provide a comparison between several adaptive strategies known in the literature to compute these parameters.

In Chapter 2, we make a step further into the nondifferentiable case. First, we introduce the mathematical notions of Convex and Variational Analysis required for the subsequent discussion, including a summary of the main subdifferential calculus rules. Then we define the concept of proximity operator of a convex function and consider its main properties and examples. Afterwards, we provide a self-contained overview of proximal–gradient methods, which constitute the natural extension of gradient methods to the nondifferentiable setting and are suited for problems with a specific structure, i.e. when the objective function is given by the sum of a convex and a differentiable term. We also highlight strengths and possible limitations of these approaches. Finally, we discuss the convergence of inexact proximal–gradient methods under the assumption that the objective function satisfies the Kurdyka–Łojasiewicz property.

In Chapter 3, we focus on the description, convergence analysis and application of the VMILAn method. In the first part of the chapter, we introduce a wide class of forward–backward algorithms developed in [32], in which the notion of proximal operator is replaced by a more general tool, in order to allow the use of non Euclidean distance in the metric. In the second part, we present VMILAn as a special instance of the previous framework, detail its key features and develop the related convergence analysis. The final section of this chapter is then devoted to some numerical illustrations on three image processing applications: image deconvolution in presence of signal dependent Gaussian noise, image deblurring in presence of Cauchy noise and linear diffusion based image compression. We remark that, in these numerical experiments, we apply several of the steplength and scaling matrix selection rules discussed in Chapter 1 and 2.

In Chapter 4, we exploit the line–search based methods developed in the previous chapter for the problem of phase estimation from color images arising in DIC microscopy. We start by devising an extension of the DIC imaging model proposed in [118] to the case of RGB images. On the basis of this model, we derive the corresponding maximum likelihood function, which is highly nonconvex, and propose to regularize it by means of the HS regularizer. We study the analytical properties of the resulting objective function and prove the existence of minimum points. The two proposed line–search based algorithms are then presented. Finally, numerical comparison of the proposed algorithms with several nonlinear conjugate gradient methods is reported.

In Chapter 5, the CBGGP algorithm for differentiable optimization problems with separable constraints is proposed. The first section is concerned with the introduction of a generalized projection operator, which is slightly different from the one introduced in Chapter 3 (indeed the latter one is a special instance of the former if the functions involved are differentiable), and on which the proposed algorithm relies on. In the following section, CBGGP is introduced together with the proof of global convergence towards stationary points. The final section is devoted to the numerical application of CBGGP to blind deconvolution.

In Appendix A, a summary of the basic notions of image restoration is presented for the reader's convenience. Indeed several of the concepts introduced here are extensively used throughout the entire thesis.

Publications

Accepted Journal Articles

1. S. Bonettini, I. Loris, F. Porta, M. Prato and S. Rebegoldi. On the convergence of a line-search based proximal-gradient method for nonconvex optimization. *Inverse Problems*, 33:055005, 2017.
2. S. Bonettini, M. Prato and S. Rebegoldi. A cyclic block coordinate descent method with generalized gradient projections. *Applied Mathematics and Computation*, 286:288–300, 2016.
3. M. Prato, A. La Camera, S. Bonettini, S. Rebegoldi, M. Bertero, and P. Boccacci. A blind deconvolution method for ground-based telescopes and Fizeau interferometers. *New Astronomy*, 40:1–13, 2015.

Conference Proceedings

1. S. Rebegoldi, L. Bautista, L. Blanc-Féraud, M. Prato, L. Zanni and A. Plata. TV-regularized phase reconstruction in differential-interference-contrast (DIC) microscopy. In *AIP Conference Proceedings*, volume 1776, page 090043, 2016.
2. M. Prato, S. Bonettini, I. Loris,. F. Porta and S. Rebegoldi. On the constrained minimization of smooth Kurdyka-Łojasiewicz functions with the scaled gradient projection method. In *Journal of Physics: Conference Series*, volume 756, page 012004, 2016.
3. L. Bautista, S. Rebegoldi, L. Blanc-Féraud, M. Prato, L. Zanni and A. Plata. Phase estimation in differential-interference-contrast (DIC) microscopy. In *IEEE Proceedings of the 13th International Symposium on Biomedical Imaging (ISBI)*, pages 136–139, 2016.
4. S. Rebegoldi, S. Bonettini, and M. Prato. Application of cyclic block generalized gradient projection methods to Poisson blind deconvolution. In *Proceedings of the 23rd European Signal Processing Conference*, pages 225-229, 2015.
5. M. Prato, S. Bonettini, A. La Camera, and S. Rebegoldi. Alternating minimization for Poisson blind deconvolution in astronomy. In *Proceedings of the Inverse Problems from Theory to Applications Conference (IPTA 2014)*, pages 148–152, 2014.

Submitted Journal Articles

1. S. Rebegoldi, L. Bautista, L. Blanc-Féraud, M. Prato, L. Zanni and A. Plata. A comparison of edge–preserving approaches for differential interference contrast microscopy. *Inverse Problems*, under revision.

Notations

- $\mathbb{R}_{\geq 0} = \{x \in \mathbb{R} : \ x \geq 0\}$ and $\mathbb{R}_{>0} = \{x \in \mathbb{R} : \ x > 0\}$ are the sets of nonnegative and positive real numbers, respectively.
- $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ is the extended real numbers set.
- $\boldsymbol{e}$ and 0 denote a vector with all entries equal to 1 and 0, respectively.
- If $x, y \in \mathbb{R}^n$, then $x^T y = \sum_{i=1}^n x_i y_i$ denotes the scalar product.
- If $x, y \in \mathbb{R}^n$, then $\frac{x}{y}$ and $x \cdot y$ denote the component-wise division and product, respectively.
- If $x \in \mathbb{R}^n$, $x \geq 0 \Leftrightarrow x_i \geq 0, \ i = 1, \dots, n$. An analogous notation holds for $>$, $\leq$, $<$.
- $D \in \mathbb{R}^{m \times n}$ denotes a matrix of m rows and n columns.
- $I_n \in \mathbb{R}^{n \times n}$ denotes the $n \times n$ identity matrix.
- $\| \cdot \|$ will denote the Euclidean norm: $\|x\| = \|x\|_2 = \sqrt{x^T x}$.
- $\| \cdot \|_D$ will denote the norm induced by a symmetric positive definite matrix $D \in \mathbb{R}^{n \times n}$: $\|x\|_D = \sqrt{x^T D x}$.
- Given $\mu \geq 1$, $\mathcal{M}_\mu$ is the set of all symmetric positive definite matrices with eigenvalues contained in the interval $[\frac{1}{\mu}, \mu]$.
- Given $\rho \in \mathbb{R}_{>0}$, $B(x, \rho) = \{y \in \mathbb{R}^n : \ \|y - x\| \leq \rho\}$ is the closed ball of center x and radius ρ.
- Given $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ and $\alpha \in \mathbb{R}$, $[f = \alpha] = \{x \in \mathbb{R}^n : \ f(x) = \alpha\}$ denotes the level set of f at height α. Analogous notations are used for the lower and upper level sets $[f \leq \alpha]$, $[f < \alpha]$, $[f \geq \alpha]$, $[f > \alpha]$.
- Given $-\infty < \alpha_1 < \alpha_2 \leq +\infty$, $[\alpha_1 < f < \alpha_2] = \{x \in \mathbb{R}^n : \ \alpha_1 < f(x) < \alpha_2\}$ is the sublevel set of f at levels α_1 and α_2.

Chapter 1

Gradient methods for differentiable optimization

The purpose of this chapter is to provide an insight into one of the most studied class of numerical algorithms designed to address the following optimization problem

$$\min_{x\in\Omega} f(x) \tag{1.1}$$

where $\Omega \subset \mathbb{R}^n$ is a nonempty, closed and convex set and $f : \Omega \to \mathbb{R}$ is a continuously differentiable function over Ω. When $\Omega = \mathbb{R}^n$ and therefore no restrictions on the unknown variable x are imposed, one speaks of *unconstrained optimization*, otherwise of *constrained optimization*.

Gradient methods are by far the most standard and popular iterative schemes aimed at solving problem (1.1). These methods are characterized by a simple implementation and a low computational cost per iteration, since they only exploit first-order information of the objective function to define each iterate. Furthermore, many of these algorithms lean on the so-called *iterative descent* idea, which consists in generating a sequence of iterates $\{x^{(k)}\}_{k\in\mathbb{N}} \subseteq \Omega$ in such a way that f is decreased at each iteration. The legitimate hope is that the iterates will eventually approach a global minimum or, at least, a stationary point of f. The decrease in the objective function is imposed by moving along a *descent direction* with a sufficiently small positive *steplength*, which may vary at each iteration. Strategies based on an adaptive choice of the steplength and the parameters involved in the definition of the descent direction allow to practically improve the convergence rate of gradient methods without increasing their computational costs. When constrained optimization is considered, the additional cost of projecting the iterates on the feasible set Ω in order to preserve feasibility must be taken into account; however, if the constraints are simple, the projection can be performed by linear-time algorithms and thus without significant computational effort.

The chapter is organized as follows. Section 1.1 is devoted to the analysis of gradient methods for unconstrained optimization, with related discussions on the choice of their parameters and convergence results, whereas Section 1.2 focuses on gradient projection methods for constrained optimization, including a similar analysis to the one of Section 1.1.

1.1 Unconstrained case: gradient methods

In this section the unconstrained version of problem (1.1) is considered, where f is assumed to be continuously differentiable on $\mathbb{R}^n$ and $\Omega = \mathbb{R}^n$. We recall the following definitions and basic results.

Definition 1.1. *A point x^* is an unconstrained* global *minimum of f if*

$$f(x^*) \leq f(x), \quad \forall\, x \in \mathbb{R}^n.$$

A point x^* is an unconstrained local *minimum of f if there exists $\epsilon > 0$ such that*

$$f(x^*) \leq f(x), \quad \forall\, x : \; \|x - x^*\| \leq \epsilon.$$

Theorem 1.1. *[22, Proposition 1.1.1] Let x^* be an unconstrained local minimum of f. Then*

$$\nabla f(x^*) = 0. \tag{1.2}$$

If, in addition, f is twice continuously differentiable over an open set U containing x^*, then also the following holds

$$\nabla^2 f(x^*) \; is \; positive \; semidefinite. \tag{1.3}$$

Equations (1.2) and (1.3) are the *first* and *second order* necessary optimality conditions, respectively, for a point x^* to be a (local) minimum of f.

Theorem 1.2. *[22, Proposition 1.1.2] Let $f : \Omega \to \mathbb{R}$ be a convex function over the convex set Ω.*

(i) A local minimum of f over Ω is also a global minimum over Ω. If, in addition, f is strictly convex, then f admits at most one global minimum.

(ii) If f is differentiable and Ω is an open set, then $\nabla f(x^*) = 0$ is a necessary and sufficient condition for a vector $x^* \in \Omega$ to be a global minimum of f over Ω.

Definition 1.2. *A point $x^* \in \mathbb{R}^n$ is stationary for f if $\nabla f(x^*) = 0$.*

Definition 1.3. *A vector $d \in \mathbb{R}^n$ is a descent direction for f at the point $x \in \mathbb{R}^n$ if*

$$\nabla f(x)^T d < 0.$$

A gradient method is an iterative algorithm which, starting from an initial guess $x^{(0)} \in \mathbb{R}^n$, generates a sequence of the form

$$x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}, \quad k = 0, 1, 2, \ldots \tag{1.4}$$

where $d^{(k)}$ is a descent direction at $x^{(k)}$ and α_k is a positive parameter denominated steplength. The majority of gradient methods are also *descent algorithms*. Indeed if one considers the half line of points

$$x_\alpha = x^{(k)} + \alpha d^{(k)}, \quad \forall\ \alpha \geq 0,$$

from the first order Taylor series expansion around $x^{(k)}$ we have

$$f(x_\alpha) = f(x^{(k)}) + \alpha \nabla f(x^{(k)})^T d^{(k)} + o(\alpha).$$

For α sufficiently small, the negative term $\alpha \nabla f(x^{(k)})^T d^{(k)}$ dominates on $o(\alpha)$ and thus $f(x_\alpha)$ is smaller than $f(x^{(k)})$. Many gradient methods are then equipped with steplength rules aimed at imposing a decrease in the objective function. However there are some exceptions; see Section 1.1.2 for an insightful discussion.

In the following we will discuss some of the most commonly known choices for the descent direction $d^{(k)}$ and the steplength α_k.

1.1.1 Choice of the descent direction

The descent direction in (1.4) is usually of the form

$$d^{(k)} = -D_k^{-1} \nabla f(x^{(k)}) \tag{1.5}$$

where D_k is a positive definite symmetric matrix called *scaling matrix*. Indeed any $d^{(k)}$ defined as in (1.5) is a descent direction, since

$$\nabla f(x^{(k)})^T d^{(k)} = \nabla f(x^{(k)})^T (-D_k^{-1} \nabla f(x^{(k)})) = -\nabla f(x^{(k)})^T D_k^{-1} \nabla f(x^{(k)}) < 0$$

where the inequality follows from the positive definiteness of the matrix D_k^{-1}. We now give some classical examples of choices of the matrix D_k.

Identity matrix

$$D_k = I_n, \quad k = 0, 1, 2, \dots \tag{1.6}$$

In this case $d^{(k)} = -\nabla f(x^{(k)})$. The resulting method is the popular *steepest descent*. The name is derived from the fact that the (normalized) negative gradient direction $p^{(k)} = -\frac{\nabla f(x^{(k)})}{\|\nabla f(x^{(k)})\|}$ is the one that, among all normalized directions $d \in \mathbb{R}^n$, minimizes the slope $\nabla f(x^{(k)})^T d$ of the function $f(x^{(k)} + \alpha d)$ at $\alpha = 0$. In fact, by the Cauchy-Schwartz inequality, we have

$$\nabla f(x^{(k)})^T d \geq -\|\nabla f(x^{(k)})\| \cdot \|d\| = -\|\nabla f(x^{(k)})\|, \quad \forall\ d \in \mathbb{R}^n,\ \|d\| = 1$$

and the inequality above is attained with $d = p^{(k)}$.
As we will see in Section 1.1.2, the steepest descent often leads to slow convergence when the problem is ill-conditioned. In particular, when the level sets of the objective f are “elongated”,

the method typically zig-zags without making fast progress towards the solution.

Hessian matrix

The *Newton's method* corresponds to the following choice

$$D_k = \nabla^2 f(x^{(k)}), \quad k = 0, 1, 2, \dots \tag{1.7}$$

provided that $\nabla^2 f(x^{(k)})$ is positive definite. When $\alpha_k = 1$, the iterate $x^{(k+1)}$ given by Newton's method is exactly the minimum point of the quadratic approximation of f around the current point $x^{(k)}$, that is

$$q^{(k)}(x) = f(x^{(k)}) + \nabla f(x^{(k)})^T(x - x^{(k)}) + \frac{1}{2}(x - x^{(k)})^T \nabla^2 f(x^{(k)})(x - x^{(k)}).$$

Indeed the general Newton iteration is easily obtained by setting $\nabla q^{(k)}(x^{(k+1)}) = 0$. In contrast with the steepest descent, Newton's method converges Q-superlinearly and does not show a zig-zagging behaviour. However, fast convergence comes with the price of evaluating the Hessian of f at each iteration, which can be computationally expensive or sometimes even impracticable.

Diagonal matrix

A convenient alternative to the previous approaches is to adopt a *diagonal* scaling matrix

$$D_k = \begin{pmatrix} d_1^{(k)} & 0 & 0 & \cdots & 0 \\ 0 & d_2^{(k)} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & d_n^{(k)} \end{pmatrix}, \quad k = 0, 1, 2, \dots \tag{1.8}$$

where $d_i^{(k)} > 0$, $i = 1, \dots, n$, in order to ensure the positive definiteness of D_k, and $d_i^{(k)}$ approximates the i-th second partial derivative of f

$$d_i^{(k)} \approx \frac{\partial^2 f(x^{(k)})}{\partial x_i^2}, \quad i = 1, \dots, n. \tag{1.9}$$

The resulting method is denominated *diagonally scaled steepest descent* and can be seen as a diagonal approximation of the Newton's method. It is evident that an appropriate choice of a diagonal scaling matrix is closely connected to the structure of the objective function f.

1.1.2 Choice of the steplength

Classical rules

The most classical steplength selection rule was first considered in [40] and consists in minimizing the objective function along the descent direction $d^{(k)}$, that is

$$\alpha_k^{SD} = \operatorname*{argmin}_{\alpha \geq 0} f(x^{(k)} + \alpha d^{(k)}). \tag{1.10}$$

Equation (1.10) is called the *minimization rule* and guarantees the monotonicity of the sequence $\{f(x^{(k)})\}_{k\in\mathbb{N}}$. When $d^{(k)} = -\nabla f(x^{(k)})$ and α_k is chosen as in (1.10), we obtain the Cauchy steepest descent method, which sometimes is simply denominated steepest descent.

A more easily implementable variant of rule (1.10) is the *limited minimization rule*, according to which the steplength α_k is chosen by minimizing $f(x^{(k)}+\alpha d^{(k)})$ on a closed and bounded interval, i.e.

$$\alpha_k^{SD} = \operatorname*{argmin}_{\alpha\in[0,s]} f(x^{(k)} + \alpha d^{(k)}) \tag{1.11}$$

where $s > 0$ is a fixed scalar. Both rules (1.10) and (1.11) are implemented with the aid of one-dimensional linesearch algorithms (see e.g. [22, Appendix C]), which often require a considerable computational effort. For that reason, it is usually preferred to successively reduce the steplength until a certain sufficient decrease condition is satisfied. This is the case of the well-known *Armijo rule* [22, 104], which is reported in Algorithm 1.

Algorithm 1 Armijo linesearch algorithm

```
Let $\{x^{(k)}\}_{k\in\mathbb{N}}$ be a sequence of points in $\mathbb{R}^n$. Choose some $\delta, \beta \in (0, 1)$, $\alpha > 0$.
1. Set $\alpha_k = \alpha$. Let $d^{(k)}$ be a descent direction at $x^{(k)}$.
2. IF
       $f(x^{(k)} + \alpha_k d^{(k)}) \le f(x^{(k)}) + \beta\alpha_k \nabla f(x^{(k)})^T d^{(k)}$      (1.12)
   THEN go to step 3.
   ELSE set $\alpha_k = \delta\alpha_k$ and go to step 2.
3. END
```

According to Algorithm 1, the steplength is set equal to $\alpha_k = \delta^{m_k}\alpha$, where m_k is the first nonnegative integer such that (1.12) is satisfied.

The main convergence result for the three previously described rules is the stationarity of the limit points of the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$, provided that the directions $d^{(k)}$ tend not to be asymptotically orthogonal to the gradient $\nabla f(x^{(k)})$.

Theorem 1.3. *[22, Proposition 1.2.1] Let $\{x^{(k)}\}_{k\in\mathbb{N}}$ be a sequence generated by a gradient method $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$ and suppose that $\{d^{(k)}\}_{k\in\mathbb{N}}$ is gradient related to $\{x^{(k)}\}_{k\in\mathbb{N}}$, i.e. for any $K \subseteq \mathbb{N}$ such that $\{x^{(k)}\}_{k\in K}$ converges to a nonstationary point, the corresponding subsequence $\{d^{(k)}\}_{k\in K}$ is bounded and satisfies*

$$\limsup_{k\to\infty,\ k\in K} \nabla f(x^{(k)})^T d^{(k)} < 0. \tag{1.13}$$

If α_k is chosen according to one of the rules (1.10), (1.11) *or* (1.12), *then every limit point of $\{x^{(k)}\}_{k\in\mathbb{N}}$ is stationary for f.*

Remark 1.1. Condition (1.13) is rather technical and difficult to verify in practice. However, such a condition is satisfied when the eigenvalues of the matrices $(D_k)^{-1}$ are bounded above and away from zero, namely if there are two positive scalars m and M such that

$$m\|x\|^2 \leq \|x\|_{D_k^{-1}} \leq M\|x\|^2. \tag{1.14}$$

If the scaling matrices are diagonal, the boundedness of the eigenvalues can be easily enforced by setting $(D_k)_{ii}^{-1} = \max\{\min\{d_i^{(k)}, \mu\}, \frac{1}{\mu}\}$, where $\mu > 0$ is a prefixed scalar.

Although Theorem 1.3 is quite general and applies to a certain number of gradient methods, it does not reveal much about their speed of convergence. To this purpose, it may be convenient to assume that the objective function is strictly convex quadratic, i.e. of the form

$$f(x) = \frac{1}{2}x^T Ax - b^T x \tag{1.15}$$

where $A \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix. In this case, since $\nabla f(x) = Ax - b$ and Theorem 1.2 holds, there exists a unique global minimum $x^* = A^{-1}b$ and any of the gradient methods previously described converges to x^*, as stated in the following result.

Theorem 1.4. *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is of the form* (1.15) *with A symmetric positive definite, denote with x^* the unique minimum point of f and let $\{x^{(k)}\}_{k\in\mathbb{N}}$ be a sequence satisfying the hypotheses of Theorem 1.3. Then $x^{(k)}$ converges to x^*.*

Proof. Since f is convex quadratic, it is coercive, namely $\lim_{\|x\|\to+\infty} f(x) = +\infty$. This implies that the set $\Omega_0 = \{x \in \mathbb{R}^n : \ f(x) \leq f(x^{(0)})\}$ is bounded since, otherwise, there would be a sequence $\{z^{(k)}\}_{k\in\mathbb{N}}$ such that $\|z^{(k)}\| \to +\infty$ and $f(z^{(k)}) \leq f(x^{(0)})$ for all k, which is absurd given the coercivity of f. By observing that the sequence $\{f(x^{(k)})\}_{k\in\mathbb{N}}$ is nonincreasing when α_k is computed by means of one of the formulae (1.10)-(1.12), we obtain that $\{x^{(k)}\}_{k\in\mathbb{N}} \subseteq \Omega_0$ and thus $\{x^{(k)}\}_{k\in\mathbb{N}}$ admits at least one limit point $\bar{x}$. Theorem 1.3 ensures that $\bar{x}$ is stationary for f and, because of Theorem 1.2, that means $\bar{x}$ is the unique limit point and $\bar{x} = x^*$. □

In the quadratic case, it is then reasonable to question whether a gradient method converges fast or not to the solution. We now report an important result concerning the convergence rate of the Cauchy steepest descent method.

Theorem 1.5. *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is of the form* (1.15) *with A symmetric positive definite, and let $\{x^{(k)}\}_{k\in\mathbb{N}}$ be generated by the Cauchy steepest descent method. If λ_1 and λ_n are the smallest and biggest eigenvalue of A, respectively, then the following inequality holds*

$$f(x^{(k+1)}) - f(x^*) \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 \left(f(x^{(k)}) - f(x^*)\right) \tag{1.16}$$

or equivalently

$$\|x^{(k+1)} - x^*\|_A \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right) \|x^{(k)} - x^*\|_A. \tag{1.17}$$

The proof of Theorem 1.5 can be found in [22, Proposition 1.3.1] or in [96]. Inequalities (1.16)-(1.17) show that the function values and the iterates converge Q-linearly to the optimal value and the minimum point of f, respectively. On one hand, those inequalities imply that, if all eigenvalues of the matrix A are equal, i.e. A is a multiple of the identity matrix, then convergence is achieved in one iteration. On the other hand, as the condition number $\kappa(A) = \lambda_n/\lambda_1$ of the Hessian matrix increases, the factor $(\lambda_n - \lambda_1)/(\lambda_n + \lambda_1)$ gets close to 1, thus suggesting that the algorithm may converge slowly. Furthermore, it can be shown that inequalities (1.16)-(1.17) are sharp, in the sense that, given any quadratic objective function, there is an initial guess $x^{(0)}$ such that those inequalities hold as an equation for all k [22, Figure 1.3.2]. In other words, there is no chance to obtain a better convergence rate result for the Cauchy steepest descent. Finally, we remark that a similar result to Theorem 1.5 is obtained also for the non-quadratic case, by assuming that the method converges to a minimum point in which the Hessian is positive definite [104, Theorem 3.4].

Another meaningful result about the asymptotic behaviour of the steepest descent is the following.

Theorem 1.6. *[3, Theorem 4], [103, Proposition 3.1] Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is of the form (1.15) with A symmetric positive definite, and let $\{x^{(k)}\}_{k\in\mathbb{N}}$ be generated by the Cauchy steepest descent method. Let $\{d_i\}_{i=1}^n$ be a basis of eigenvectors of A, and assume that the corresponding eigenvalues $0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{n-1} \leq \lambda_n$ are such that $\lambda_1 < \lambda_2$ and $\lambda_{n-1} < \lambda_n$. For all k, define $\mu_i^{(k)} \in \mathbb{R}$, $i = 1, \ldots, n$ as the components of $g^{(k)} = \nabla f(x^{(k)})$ w.r.t. the eigenvectors d_i, namely*

$$g^{(k)} = \sum_{i=1}^{n} \mu_i^{(k)} d_i. \tag{1.18}$$

and suppose that $\mu_1^{(0)} \neq 0$ and $\mu_n^{(0)} \neq 0$. Then

(i) the following limits hold

$$\lim_{k\to\infty} \frac{(\mu_i^{(2k)})^2}{\sum_{j=1}^n (\mu_j^{(2k)})^2} = \begin{cases} \frac{1}{1+c^2}, & \text{if } i = 1 \\ 0, & \text{if } i = 2, \ldots, n-1 \\ \frac{c^2}{1+c^2}, & \text{if } i = n \end{cases}$$

$$\lim_{k\to\infty} \frac{(\mu_i^{(2k+1)})^2}{\sum_{j=1}^n (\mu_j^{(2k+1)})^2} = \begin{cases} \frac{c^2}{1+c^2}, & \text{if } i = 1 \\ 0, & \text{if } i = 2, \ldots, n-1 \\ \frac{1}{1+c^2}, & \text{if } i = n \end{cases}$$

where $c = \lim_{k\to+\infty} \mu_n^{(2k)}/\mu_1^{(2k)} = -\lim_{k\to+\infty} \mu_1^{(2k+1)}/\mu_n^{(2k+1)}$;

(ii) the components $\mu_1^{(2k)}$, $\mu_n^{(2k)}$, $\mu_1^{(2k+1)}$, $\mu_n^{(2k+1)}$ have fixed signs for large k.

In other words, the even and odd normalized gradients converge to two distinct points

$$\lim_{k\to+\infty} \frac{g^{(2k)}}{\|g^{(2k)}\|} = \bar{d} \qquad\qquad \lim_{k\to+\infty} \frac{g^{(2k+1)}}{\|g^{(2k+1)}\|} = \hat{d},$$

and the steepest descent eventually performs its search in the 2D space spanned by d_1 and d_n, which explains the well-known zigzagging behaviour of the method.

Barzilai-Borwein rules

The steplength rules described so far rely on the monotonicity of the function values in order to ensure global convergence of the algorithm. A totally different approach is adopted by the two well-known *Barzilai-Borwein* (BB) rules [11]. In particular the BB steplengths arise from the approximation of the Hessian $\nabla^2 f(x^{(k)})$ with the diagonal matrix $B(\alpha_k) = (\alpha_k I_n)^{-1}$, which is forced to assume one of the following quasi-Newton properties:

$$\alpha_k^{BB1} = \underset{\alpha\in\mathbb{R}}{\operatorname{argmin}} \|B(\alpha)s^{(k-1)} - y^{(k-1)}\| \tag{1.19}$$

$$\alpha_k^{BB2} = \underset{\alpha\in\mathbb{R}}{\operatorname{argmin}} \|s^{(k-1)} - B(\alpha)^{-1}y^{(k-1)}\|, \tag{1.20}$$

where $s^{(k-1)} = x^{(k)} - x^{(k-1)}$ and $y^{(k-1)} = \nabla f(x^{(k)}) - \nabla f(x^{(k-1)})$. The resulting values are

$$\alpha_k^{BB1} = \frac{{s^{(k-1)}}^T s^{(k-1)}}{{s^{(k-1)}}^T y^{(k-1)}} \qquad ; \qquad \alpha_k^{BB2} = \frac{{s^{(k-1)}}^T y^{(k-1)}}{{y^{(k-1)}}^T y^{(k-1)}}. \tag{1.21}$$

We remark that the BB rules were first given in the context of quadratic objective functions. In this case, by observing again that $\nabla f(x) = Ax - b$, it is true that

$$As^{(k-1)} = y^{(k-1)} \;\Rightarrow\; \|As^{(k-1)} - y^{(k-1)}\| = \|s^{(k-1)} - A^{-1}y^{(k-1)}\| = 0. \tag{1.22}$$

Since A is the Hessian of (1.15), relation (1.22) clarifies why formulae (1.19)-(1.20) are used to obtain an approximation of the Hessian matrix.

Numerical experience shows that the BB rules and their modifications are able to greatly speed up the slow convergence exhibited by the Cauchy steepest descent method, both in the quadratic [11, 67] and non-quadratic case [120]. Indeed, Barzilai and Borwein [11] were able to prove the R-superlinear convergence of the steepest descent method equipped with one of the steplength rule in (1.21) for two-dimensional strictly convex quadratic functions. Furthermore, Raydan [120] established global convergence of the BB methods for the strictly convex quadratic case with any number of variables and, in the same setting, Dai and Liao [56] showed the expected R-linear convergence result. However, these two last results hold also for the Cauchy steepest descent and do not explain why the BB methods are much more effective in practice. It would be nice to extend the R-superlinear convergence result proved in two dimensions to the $n-$dimensional case, but that seems unlikely to occur in the presence of round-off errors [62].

An explanation of the good behaviour of the BB methods is given in [62, 64] for a quadratic objective function; in this case, if a steepest descent method is considered, we can rewrite the gradient $g^{(k)} = \nabla f(x^{(k)})$ as follows

$$\begin{aligned} g^{(k)} = Ax^{(k)} - b &= Ax^{(k-1)} - b - \alpha_{k-1} A g^{(k-1)} \\ &= (I_n - \alpha_{k-1} A) g^{(k-1)}. \end{aligned} \tag{1.23}$$

Applying iteratively (1.23) yields the following relation

$$g^{(k)} = \left(\prod_{j=0}^{k-1} (I_n - \alpha_j A) \right) g^{(0)}. \tag{1.24}$$

Let $\{d_i\}_{i=1}^n$ be a basis of eigenvectors of A and $0 < \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_{n-1} < \lambda_n$ the corresponding eigenvalues. Then the vector $g^{(0)}$ may be represented in the form $g^{(0)} = \sum_{i=1}^n \mu_i^{(0)} d_i$ with $\mu_i^{(0)} \in \mathbb{R}$, $i = 1, \ldots, n$, and equation (1.24) becomes

$$g^{(k)} = \sum_{i=1}^n \mu_i^{(0)} \left(\prod_{j=0}^{k-1} (I_n - \alpha_j A) \right) d_i. \tag{1.25}$$

Finally, by comparing (1.18) with (1.25), we deduce the following relation

$$\mu_i^{(k)} = \mu_i^{(0)} \left(\prod_{j=0}^{k-1} (1 - \alpha_j \lambda_i) \right) = \mu_i^{(k-1)} \left(1 - \alpha_{k-1} \lambda_i\right). \tag{1.26}$$

From the recurrence (1.26), two fundamental facts may be deduced:

- if at the $(k-1)$-th iteration $\mu_i^{(k-1)} = 0$, then $\mu_i^{(h)} = 0$ for all $h \geq k$;
- if at the $(k-1)$-th iteration $\alpha_{k-1} = 1/\lambda_i$, then $\mu_i^{(k)} = 0$.

This means that if the first n steps of the steepest descent method are defined by setting

$$\alpha_k = \frac{1}{\lambda_k}, \quad k = 1, \ldots, n$$

then $g^{(n)} = 0$ and the method converges in (at most) n steps. Therefore, it seems desirable that the steplength α_k approximates the reciprocal of some eigenvalue of the Hessian matrix at each iteration. Since the eigenvalues of A are usually not available, one might approximate them with the Rayleigh quotients of the matrix A, which are defined as

$$R_A(x) = \frac{x^T A x}{\|x\|^2}, \quad \forall\, x \in \mathbb{R}^n \setminus \{0\}. \tag{1.27}$$

Such an approximation is reasonable, since any eigenvalue of A is a Rayleigh quotient in which x is the corresponding eigenvector and, in addition, the minimum and maximum value of $R_A(x)$ over x coincide with the minimum and maximum eigenvalue of A, respectively:

$$\lambda_1 = \min_{\substack{x\in\mathbb{R}^n \\ x\neq 0}} R_A(x) = R_A(d_1) \tag{1.28}$$

$$\lambda_n = \max_{\substack{x\in\mathbb{R}^n \\ x\neq 0}} R_A(x) = R_A(d_n) \tag{1.29}$$

The next result shows that both BB steplengths can be seen as approximations of the reciprocals of the eigenvalues of A of the form (1.27).

Proposition 1.1. *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is defined as in (1.15) and let $\{x^{(k)}\}_{k\in\mathbb{N}}$ be generated by a gradient method of the form $x^{(k+1)} = x^{(k)} - \alpha_k \nabla f(x^{(k)})$. Then the BB rules can be rewritten as follows*

$$\alpha_k^{BB1} = \frac{{g^{(k-1)}}^T g^{(k-1)}}{{g^{(k-1)}}^T A g^{(k-1)}} = R_A^{-1}(g^{(k-1)}) \tag{1.30}$$

$$\alpha_k^{BB2} = \frac{{g^{(k-1)}}^T A g^{(k-1)}}{{g^{(k-1)}}^T A^2 g^{(k-1)}} = R_A^{-1}(A^{\frac{1}{2}} g^{(k-1)}) \tag{1.31}$$

where $g^{(k-1)} = \nabla f(x^{(k-1)})$. Furthermore, if λ_1 and λ_n are the smallest and biggest eigenvalue of A, respectively, then the following property holds

$$\frac{1}{\lambda_n} \leq \alpha_k^{BB2} \leq \alpha_k^{BB1} \leq \frac{1}{\lambda_1}. \tag{1.32}$$

Proof. Combining (1.22) with relation $s^{(k-1)} = -\alpha_k g^{(k-1)}$ yields $y^{(k-1)} = -\alpha_k A g^{(k-1)}$. By replacing these two relations in α_k^{BB1} we obtain

$$\alpha_k^{BB1} = \frac{\left(-\alpha_k g^{(k-1)}\right)^T \left(-\alpha_k g^{(k-1)}\right)}{\left(-\alpha_k g^{(k-1)}\right)^T \left(-\alpha_k A g^{(k-1)}\right)} = \frac{{g^{(k-1)}}^T g^{(k-1)}}{{g^{(k-1)}}^T A g^{(k-1)}}. \tag{1.33}$$

A similar reasoning can be done for α_k^{BB2}, thus proving (1.31). Direct use of the Cauchy-Schwartz implies the following inequality

$${g^{(k-1)}}^T A g^{(k-1)} \leq \sqrt{{g^{(k-1)}}^T g^{(k-1)}} \sqrt{{g^{(k-1)}}^T A^2 g^{(k-1)}}. \tag{1.34}$$

Taking squares of both sides of (1.34) and dividing it by $({g^{(k-1)}}^T A g^{(k-1)}) \cdot ({g^{(k-1)}}^T A^2 g^{(k-1)})$ yields the inequality $\alpha_k^{BB2} \leq \alpha_k^{BB1}$ in (1.32). Finally, the inequalities $\alpha_k^{BB2} \geq 1/\lambda_n$ and $\alpha_k^{BB1} \leq 1/\lambda_1$ follow from the extremal properties of the Rayleigh quotient (1.28)-(1.29). □

Remark 1.2. If f is non-quadratic and ${s^{(k-1)}}^T y^{(k-1)} > 0$, the property $\alpha_k^{BB2} \leq \alpha_k^{BB1}$ still holds. Furthermore, by the mean-value theorem of integral calculus it follows that

$$y^{(k-1)} = \left(\int_0^1 \nabla^2 f(x^{(k-1)} + ts^{(k-1)})dt \right) s^{(k-1)}$$

and hence both steplengths (1.21) define the inverse of a Rayleigh quotient relative to the average Hessian matrix $\int_0^1 \nabla^2 f(x^{(k-1)} + ts^{(k-1)})dt$.

It should be noted that also the Cauchy steplength (1.10) can be seen as the reciprocal of a Rayleigh quotient. In fact, by computing the derivative of the quadratic function w.r.t. α

$$\frac{d}{d\alpha} f(x^{(k)} - \alpha g^{(k)}) = -{g^{(k)}}^T \left(A(x^{(k)} - \alpha g^{(k)}) - b \right) = -{g^{(k)}}^T g^{(k)} + \alpha {g^{(k)}}^T A g^{(k)}.$$

and setting it to 0, we obtain

$$\alpha_k^{SD} = \frac{{g^{(k)}}^T g^{(k)}}{{g^{(k)}}^T A g^{(k)}} = R_A^{-1}(g^{(k)}). \tag{1.35}$$

Nevertheless, there is some evidence that the eigenvalues approximations provided by the sequence $\{1/\alpha_k^{BB1}\}_{k\in\mathbb{N}}$ are much better than the ones given by the Cauchy optimal choice [64, 72]. In fact, from the recurrence (1.26), we observe that

$$\alpha_k \approx \frac{1}{\lambda_i} \quad \Rightarrow \quad \begin{cases} |\mu_i^{(k)}| \ll |\mu_i^{(k-1)}| \\ |\mu_j^{(k)}| < |\mu_j^{(k-1)}|, & \text{if } j < i \\ |\mu_j^{(k)}| > |\mu_j^{(k-1)}|, & \text{if } j > i, \ \lambda_j > 2\lambda_i. \end{cases} \tag{1.36}$$

Thus, small steplengths α_k (close to $1/\lambda_n$) tend to decrease a large number of eigencomponents, with negligible reduction of those corresponding to small eigenvalues. These latter ones can be significantly reduced by using large steplengths, which however may cause an increase in the eigencomponents corresponding to the dominating eigenvalues and thus foster non-monotonic behaviour, both for the gradient norm and the function value. For that reason, it is likely that the Cauchy steplengths α_k^{SD} tend to be small in order to ensure the expected monotonic behaviour, whereas the reciprocals of the BB steplengths $1/\alpha_k^{BB1}$ are allowed to sweep the spectrum of A, with the result of forcing each component $\mu_i^{(k)}$ to go to zero.

Another consideration that can be deduced from (1.36) is that some balance between large and small steplengths is essential in order to devise effective gradient methods. This basic idea has given rise to novel steplength selection rules, based on the alternation of the Cauchy and/or BB steplengths. Several of these methods belong to the class of Gradient Methods with Retards (GMR) [68] which, given positive integers m and q_i, $i = 1, \dots, m$, set the steplength as follows

$$\alpha_k^{GMR} = \frac{g_{\nu(k)}^T A^{\rho(k)-1} g_{\nu(k)}}{g_{\nu(k)}^T A^{\rho(k)} g_{\nu(k)}} \tag{1.37}$$

where $\nu(k) \in \{k, k-1, \dots, \max\{0, k-m\}\}$ and $\rho(k) \in \{q_1, q_2, \dots, q_m\}$. Clearly, steplengths (1.30)-(1.31)-(1.35) are all special cases of (1.37). Remarkable members of the GMR class are the Alternate Step (AS) gradient method [122, 53], in which the Cauchy and BB1 steplengths are used in turns, or the Alternate Minimization (AM) method [57], where the minimization of the objective function along the line provided by (1.10) is alternated with the one-dimensional minimization of the gradient norm. These approaches, which all rely on a prefixed alternation of the selected rules, seem to be overcome by methods where the steplengths are adaptively alternated on the basis of some switching criterion, such as the Adaptive Steepest Descent (ASD) method, the Adaptive Barzilai Borwein (ABB) method [145] and its generalizations $\mathrm{ABB}_{\min 1}$ and $\mathrm{ABB}_{\min 2}$ [67]. In particular, the $\mathrm{ABB}_{\min 1}$ method alternates the two BB rules in the following way

$$\alpha_k^{\mathrm{ABB}_{\min 1}} = \begin{cases} \min\left\{\alpha_j^{BB2} : \ j = \max\{1, k-m\}, \dots, k\right\}, & \text{if } \frac{\alpha_k^{BB2}}{\alpha_k^{BB1}} < \tau \\ \alpha_k^{BB1}, & \text{otherwise} \end{cases} \tag{1.38}$$

where m is a nonnegative integer and $\tau \in (0,1)$. Notice that, when $m = 0$, the former ABB rule is recovered. The $\mathrm{ABB}_{\min 1}$ strategy aims at generating a sequence of small steplengths with the BB2 rule so that the subsequent value generated by the BB1 rule is a suitable approximation of the reciprocal of some small eigenvalue. The switching criterion in (1.38) is based on the relation $\alpha_k^{BB2}/\alpha_k^{BB1} = \cos^2\theta_{k-1}$, where θ_{k-1} is the angle between $Ag^{(k-1)}$ and $g^{(k-1)}$, and allows to select the steplength α_k^{BB1}, which is the inverse of the Rayleigh quotient of $g^{(k-1)}$, only when $g^{(k-1)}$ itself is a sufficiently good approximation of an eigenvector. From the theoretical point of view, since ABB, $\mathrm{ABB}_{\min 1}$ and $\mathrm{ABB}_{\min 2}$ belong to the GMR class, their R-linear convergence can be proved exactly as in [53], whereas Q-linear convergence for the error norm of the ASD method is established in [145]. From the practical side, these methods have been shown to further accelerate the convergence of the standard BB method [67].

When the non-quadratic case is considered, the BB method needs be equipped with a linesearch strategy that allows the objective function to increase at some iterations, in order to comply with the non-monotonic behaviour of the sequence $\{f(x^{(k)})\}_{k\in\mathbb{N}}$, while still guaranteeing global convergence of the sequence. In [121] Raydan suggested to make use of the nonmonotone linesearch technique devised by Grippo, Lampariello and Lucidi in [75], which is based on a generalization of the Armijo rule (1.12). In particular, for given scalars $\beta, \delta \in (0,1)$, $\epsilon > 1$, $\gamma > 0$, and by setting

$$\alpha_k^{(0)} = \begin{cases} \alpha_k^{BB1}, & \text{if } \ \alpha_k^{BB1} \in [\frac{1}{\epsilon}, \epsilon] \\ \gamma, & \text{otherwise} \end{cases}$$

as initial guess, then the steplength α_k is chosen as $\delta^{m_k}\alpha_k^{(0)}$, where m_k is the first nonnegative integer for which

$$f(x^{(k)} + \delta^{m_k}\alpha_k^{(0)}d^{(k)}) \leq f_{max} + \beta\delta^{m_k}\alpha_k^{(0)}\nabla f(x^{(k)})^T d^{(k)}, \tag{1.39}$$

is satisfied, where $f_{max} = \max_{0 \leq j \leq \min(k, M-1)} f(x^{(k-j)})$ is the maximum value of the objective function over the last M iterations, being M a prefixed positive integer. Notice that, if M is set equal to 1, the standard Armijo rule (1.12) is recovered. The resulting scheme, which is denominated Global Barzilai and Borwein (GBB) algorithm, is globally convergent, in the sense that each limit point of its sequence is stationary for the objective function [121, Theorem 2.1].

Ritz values based rule

We conclude this section by presenting a limited-memory steplength selection rule recently proposed by Fletcher [65] in the context of steepest descent methods. The new method was first devised for the quadratic objective function (1.15) and makes use of the most recent m back gradients

$$G = \left[g^{(k-m)} \ \ldots \ g^{(k-2)} \quad g^{(k-1)} \right] \tag{1.40}$$

to define the next m steplengths α_{k+i-1}, $i = 1, \ldots, m$. If we apply iteratively (1.23) to the vector $g^{(k-i)}$ for $m - i$ times, we obtain

$$g^{(k-i)} = \left(\prod_{j=k-m}^{k-i-1} (I_n - \alpha_j A) \right) g^{(k-m)}, \quad i = 1, \ldots, m-1,$$

that is, the gradient vectors $g^{(k-i)}$, $i = 1, \ldots, m$, belong to the span of the so-called *Krylov sequence* generated from $g^{(k-m)}$

$$\left\{ g^{(k-m)}, \ Ag^{(k-m)}, \ A^2 g^{(k-m)}, \ldots, \ A^{(m-1)} g^{(k-m)} \right\}. \tag{1.41}$$

A remarkable property of this sequence is that it provides m distinct approximations of the eigenvalues of A, denominated *Ritz values*, by means of a Lanczos iterative process [73] applied to the matrix A. Such a method starts with $q_1 = g^{(k-m)}/\|g^{(k-m)}\|$ and generates an orthonormal basis $\{q_1, q_2, \ldots, q_m\}$ for the Krylov sequence (1.41). Since the columns of G belong to the Krylov sequence, we can write $G = QR$, where Q is the $n \times m$ orthogonal matrix with columns $q_1, q_2, \ldots, q_m$ and R is a $m \times m$ upper triangular matrix which is non singular, provided that the columns of G are linearly independent. The Ritz values are then given by the eigenvalues of the matrix

$$\Phi = Q^T A Q,$$

which is tridiagonal. If $m = n$, the Ritz values θ_i, $i = 1, \ldots, m$, coincide with the eigenvalues of A while, if $m = 1$, then $Q = q_1 = g^{(k-m)}/\|g^{(k-m)}\|$ and there is a unique Ritz value, i.e. the Rayleigh quotient of $g^{(k-1)}$ on which the BB method is based. For a general m, the Ritz values are contained in the spectrum of A, since each one of them can be seen as the Rayleigh quotient $\theta_i = R_A(Qy_i)$ in which y_i is an eigenvector associated to θ_i and, in addition, the smallest and biggest Ritz values converge to the minimum and maximum eigenvalue of A, respectively, as $m \to \infty$ [80].

The idea suggested by Fletcher is to divide the sequence of the steepest descent method into groups of m iterations denominated *sweeps*, and select the next m steplengths for the current sweep as the reciprocals of the m Ritz values available from the previous sweep, namely

$$x^{(k+i)} = x^{(k+i-1)} - \alpha_{k+i-1} g^{(k+i-1)}, \quad i = 1, \ldots, m \tag{1.42}$$

where $\alpha_{k+i-1} = (\theta_{k+i-1})^{-1}$. The resulting method is called Limited Memory Steepest Descent (LMSD), which is proved to be convergent in the quadratic case [65] by following the same line of proof exploited in [120] for the BB method.

We remark that the Ritz values can be computed without explicitly using the matrices A and Q. This is important not only to reduce the computational time of the LMSD method, but also to further extend the rule to the non-quadratic case, where the matrix A is not available. Indeed, by rewriting equation (1.23) as follows

$$g^{(k+1)} = g^{(k)} - \alpha_k A g^{(k)}$$

then it can be rearranged in the matrix form

$$AG = [G \quad g^{(k)}]\Gamma \tag{1.43}$$

where Γ is a $(m+1) \times m$ matrix containing the reciprocals of the corresponding last m steplengths

$$\Gamma = \begin{bmatrix} \alpha_{k-m}^{-1} & & & \\ -\alpha_{k-m}^{-1} & \ddots & & \\ & \ddots & \alpha_{k-2}^{-1} & \\ & & -\alpha_{k-2}^{-1} & \alpha_{k-1}^{-1} \\ & & & -\alpha_{k-1}^{-1} \end{bmatrix}.$$

Combining (1.43) with relation $Q = GR^{-1}$ yields

$$\Phi = Q^T AGR^{-1} = [R \quad Q^T g^{(k)}]\Gamma R^{-1}.$$

By introducing the vector $r = Q^T g^{(k)}$, that is the vector which solves the linear system $R^T r = G^T g^{(k)}$, we obtain

$$\Phi = [R \quad r]\Gamma R^{-1}. \tag{1.44}$$

Then one needs to determine the Cholesky factorization $G^T G = R^T R$ and solve the upper triangular linear system $R^T r = G^T g^{(k)}$ before computing the tridiagonal matrix Φ via equation (1.44), in which Q does not appear.

For a general objective function, Φ is upper Hessenberg and the Ritz-like values are obtained by computing the eigenvalues of a symmetric and tridiagonal approximation $\widetilde{\Phi}$ of Φ defined as

$$\widetilde{\Phi} = \operatorname{diag}(\Phi) + \operatorname{tril}(\Phi, -1) + \operatorname{tril}(\Phi, -1)^T, \tag{1.45}$$

where diag($\cdot$) and tril($\cdot, -1$) denote the diagonal and the strictly lower triangular parts of a matrix, respectively. Possible negative eigenvalues of the resulting matrix are discarded before using this set of steplengths for the next iterations. Several numerical experiments [65], for both quadratic and nonquadratic test problems, demonstrate that the LMSD method outperforms the standard Barzilai Borwein scheme, as well as being competitive with other state-of-the-art methods, such as the BFGS method or the nonlinear Conjugate Gradient (CG) methods.

1.2 Constrained case: gradient projection methods

We now turn to the original constrained minimization problem (1.1) and recall the following basic definitions.

Definition 1.4. *A vector $x^* \in \Omega$ is a stationary point of f over Ω if*

$$\nabla f(x^*)^T (y - x^*) \geq 0, \quad \forall\ y \in \Omega. \tag{1.46}$$

Definition 1.5. *Let D be a symmetric positive definite $n \times n$ matrix. The projection operator $P_{\Omega,D} : \mathbb{R}^n \to \Omega$ is defined as*

$$P_{\Omega,D}(x) = \arg\min_{y\in\Omega} \|y - x\|_D = \operatorname*{argmin}_{y\in\Omega} \left(\phi(y) \equiv \frac{1}{2} y^T D y - y^T D x \right). \tag{1.47}$$

From Definition 1.4 and the strict convexity of the function ϕ introduced in (1.47), we deduce that an equivalent definition of $P_{\Omega,D}$ is the following

$$\left(P_{\Omega,D}(x) - x\right)^T D \left(P_{\Omega,D}(x) - y\right) \leq 0, \quad \forall\ y \in \Omega. \tag{1.48}$$

Lemma 1.1. *Let $x^* \in \Omega$ and, for any positive scalar α and symmetric positive definite matrix D, define $d^* = P_{\Omega,D}(x^* - \alpha D^{-1} \nabla f(x^*)) - x^*$.*

(i) x^* is a stationary point of f if and only if $d^* = 0$;

(ii) if $d^* \neq 0$, then d^* is a descent direction for f at x^*, that is $\nabla f(x^*)^T d^* < 0$.

Proof. (i) Assume that $x^* = P_{\Omega,D}(x^* - \alpha D^{-1} \nabla f(x^*))$. From (1.48) we have

$$(x^* - x^* + \alpha D^{-1} \nabla f(x^*))^T D (x^* - x) \leq 0, \quad \forall\ x \in \Omega$$

which implies the stationarity condition (1.46).
Conversely, let $x^* \in \Omega$ be a stationary point for f and, by contradiction, suppose that $\bar{x} = P_{\Omega,D}(x^* - \alpha D^{-1} \nabla f(x^*))$ with $\bar{x} \neq x^*$. It follows again from (1.48) that

$$(\bar{x} - x^* + \alpha D^{-1} \nabla f(x^*))^T D (\bar{x} - x^*) \leq 0$$

or equivalently

$$\|\bar{x} - x^*\|_D^2 + \alpha \nabla f(x^*)^T (\bar{x} - x^*) \leq 0.$$

The previous inequality yields

$$\nabla f(x^*)^T(\bar{x} - x^*) \leq -\frac{\|\bar{x} - x^*\|_D^2}{\alpha} < 0$$

which is contrast with the stationarity assumption on x^*.
(ii) From inequality (1.48) with $x = x^* - \alpha D^{-1}\nabla f(x^*)$ and $y = x^*$, it follows that

$$(d^* + \alpha D^{-1}\nabla f(x^*))^T D d^* \leq 0$$

which implies that

$$\nabla f(x^*)^T d^* \leq -\frac{\|d^*\|_D^2}{\alpha} < 0.$$

□

1.2.1 Classical gradient projection approaches

A simple and well studied algorithm for the solution of the constrained optimization problem (1.1) is the Gradient Projection (GP) method, whose general iteration is given by

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \lambda_k d^{(k)} = \\ &= x^{(k)} + \lambda_k \left(P_\Omega(x^{(k)} - \alpha_k \nabla f(x^{(k)})) - x^{(k)} \right), \end{aligned} \qquad (1.49)$$

where $\lambda_k \in (0,1]$ is the linesearch parameter, α_k is a positive steplength and $P_\Omega = P_{\Omega,I_n}$ is the projection operator induced by the matrix I_n, i.e. the standard Euclidean projection. Note that, because of Lemma 1.1, $d^{(k)}$ is a descent direction at point $x^{(k)}$ whenever $d^{(k)} \neq 0$, otherwise $x^{(k)}$ is a stationary point for f.

Two fundamental approaches arise in the context of gradient projection methods. On one hand, there is the class of the gradient projection methods with linesearch performed *along the arc* [22, 21], in which $\lambda_k \equiv 1$ and the steplength α_k is determined by successive reductions until an Armijo-like inequality is satisfied. In other words, if we define the *projection arc* as the set

$$\{x^{(k)}(\alpha) : \ \alpha > 0\}$$

where $x^{(k)}(\alpha) = P_\Omega(x^{(k)} - \alpha \nabla f(x^{(k)}))$, and we fix scalars $\beta, \delta \in (0,1)$, $\bar{\alpha} > 0$, then the next iterate is chosen as $x^{(k+1)} = x^{(k)}(\alpha_k)$ with $\alpha_k = \delta^{m_k}\bar{\alpha}$, where m_k is the first nonnegative integer for which

$$f(x^{(k)}(\delta^{m_k}\bar{\alpha})) \leq f(x^{(k)}) + \beta \nabla f(x^{(k)})^T (x^{(k)}(\delta^{m_k}\bar{\alpha}) - x^{(k)}). \qquad (1.50)$$

One main disadvantage of this strategy is that a projection onto the feasible set Ω must be performed for each trial point $x^{(k)}(\delta^{m_k}\bar{\alpha})$, which could become computationally too expensive if the linesearch requires many successive reductions. On the other hand, the *along the feasible direction* approach determines the next iterate as $x^{(k+1)} = x^{(k)} + \lambda_k d^{(k)}$, where λ_k is determined by means of a backtracking loop where the Armijo rule (1.12) or its nonmonotone version (1.39)

Algorithm 2 Scaled Gradient Projection (SGP) method

```
Choose the starting point $x^{(0)} \in \Omega$, set the parameters $\beta, \delta \in (0,1)$, $0 < \alpha_{min} < \alpha_{max}$, $\mu \geq 1$
and fix a positive integer $M = 1$.
FOR $k = 0, 1, 2, \ldots$
    STEP 1. Choose $\alpha_k \in [\alpha_{min}, \alpha_{max}]$, $\mu_k \leq \mu$ and the scaling matrix $D_k \in \mathcal{M}_{\mu_k}$.
    STEP 2. Compute the projection $y^{(k)} = P_{\Omega, D_k}(x^{(k)} - \alpha_k D_k^{-1} \nabla f(x^{(k)}))$;
            if $y^{(k)} = x^{(k)}$, then $x^{(k)}$ is a stationary point and SGP stops.
    STEP 3. Define the descent direction $d^{(k)} = y^{(k)} - x^{(k)}$.
    STEP 4. Set $\lambda_k = 1$ and $f_{max} = \max_{0 \leq j \leq \min(k, M-1)} f(x^{(k-j)})$.
    STEP 5. Backtracking loop:
            IF $f(x^{(k)} + \lambda_k d^{(k)}) \leq f_{max} + \beta \lambda_k \nabla f(x^{(k)})^T d^{(k)}$ THEN
                go to STEP 6
            ELSE
                set $\lambda_k = \delta \lambda_k$ and go to STEP 5.
            ENDIF
    STEP 6. Set $x^{(k+1)} = x^{(k)} + \lambda_k d^{(k)}$.
END
```

is imposed. Unlike the along the arc methods, here the projection is computed only once at each iteration.

The stationarity of the limit points of the sequences generated by both approaches is proved in [22, Proposition 2.3.1, Proposition 2.3.3]. Furthermore, when the objective function is convex and admits at least one minimum point, convergence of the whole sequence to a solution of problem (1.1) is established for both classes of methods in [86]. However, the GP method is known for being quite slow in practice, which is why several variants of such methods have been proposed in the last years [24, 25, 54, 37] in order to accelerate its convergence. In the following, we will deepen the analysis of one of these variants [37].

1.2.2 The Scaled Gradient Projection (SGP) method

The Scaled Gradient Projection (SGP) method [37] can be considered as an extension of the GP method (1.49) and is based on the following iteration

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \lambda_k d^{(k)} = \\ &= x^{(k)} + \lambda_k \left(P_{\Omega, D_k}(x^{(k)} - \alpha_k D_k^{-1} \nabla f(x^{(k)})) - x^{(k)} \right), \end{aligned} \tag{1.51}$$

where

- α_k is a positive steplength chosen in the bounded interval $[\alpha_{min}, \alpha_{max}]$;
- D_k is a symmetric positive definite matrix whose eigenvalues lie in the bounded interval $[\frac{1}{\mu}, \mu]$ with $\mu \geq 1$;
- the linesearch parameter $\lambda_k \in (0, 1]$ is determined along the feasible direction by imposing the nonmonotone Armijo rule (1.39).

The SGP method is described in its entireness in Algorithm 2 and its main convergence result is reported in Theorem 1.7.

Theorem 1.7. *[37, Theorem 2.1] Assume that the level set $\Omega_0 = \{x \in \Omega : f(x) \leq f(x^{(0)})\}$ is bounded. Every limit point of the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ generated by the SGP algorithm is a stationary point of* (1.1)*.*

Recently, convergence of the SGP sequence to a minimum point of f was proved in the convex case [35] by extending the result in [86] under the assumption that the scaling matrices D_k asymptotically reduce to the identity matrix. Such a requirement is expressed in terms of the bounds of the eigenvalues $\{\mu_k\}_{k\in\mathbb{N}}$, as better specified in the following result.

Theorem 1.8. *[35, Theorem 3.1] Assume that the objective function in* (1.1) *is convex and that a minimum point x^* over Ω exists. Let $\{x^{(k)}\}_{k\in\mathbb{N}}$ be the sequence generated by SGP where $D_k \in \mathcal{M}_{\mu_k}$ and μ_k is such that*

$$\mu_k^2 = 1 + \zeta_k, \quad \zeta_k \geq 0, \ \sum_{k=0}^{\infty} \zeta_k < \infty.$$

Then the following facts hold:

(i) the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ converges to a solution of (1.1)*;*

(ii) if f has a Lipschitz continuous gradient on Ω, then

$$f(x^{(k+1)}) - f(x^*) = \mathcal{O}\left(\frac{1}{k}\right).$$

The strength of SGP lies in its variable parameters α_k and D_k, which can be appropriately chosen, by means of adaptive strategies, in order to improve the algorithmic performances. Indeed, numerical experience in several image reconstruction problems arising in microscopy and astronomy [16, 31, 34, 94, 107, 112] has demonstrated the validity of SGP when both parameters α_k and D_k are selected in a variable and adaptive way at each iteration. In the following, we will devise some convenient updating rules for the SGP framework.

Scaling matrix choice

A clever way to determine the scaling matrix D_k, which has been extensively exploited in the aforementioned works, is provided in [89, 90] when the solution of problem (1.1) is forced to be nonnegative in each component, namely $\Omega \subseteq \{x \in \mathbb{R}^n : \ x \geq 0\}$. Such a technique is based on the following decomposition of the gradient

$$\nabla f(x) = V(x) - U(x), \quad V(x) > 0, \ U(x) \geq 0. \tag{1.52}$$

Note that this approach is widely applicable in the field of image reconstruction, in which a natural decomposition of the gradient of the form (1.52) can be found for the majority of the adopted models (see Chapter 4 for examples).

Let $x^* \in \Omega$ be a solution of problem (1.1), then x^* must comply with the Karush-Kuhn-Tucker (KKT) conditions

$$\nabla f(x^*) - \lambda = 0, \quad x^* \geq 0, \quad \lambda \geq 0, \quad x_i^* \lambda_i = 0, \quad i = 1, \dots, n \tag{1.53}$$

where $\lambda \in \mathbb{R}^n$ are the Lagrange multipliers. This implies that

$$x_i^* \nabla f(x_i^*) = 0, \quad i = 1, \dots, n. \tag{1.54}$$

On the basis of the decomposition (1.52), the n nonlinear equations (1.54) can also be rewritten as the vectorial fixed point equation

$$x^* = x^* \cdot \frac{U(x^*)}{V(x^*)}.$$

By applying the method of successive approximations, fixed an initial guess $x^{(0)} > 0$, we get the following iterative algorithm

$$x^{(k+1)} = x^{(k)} \cdot \frac{U(x^{(k)})}{V(x^{(k)})}$$

which, by recalling that $U(x^{(k)}) = V(x^{(k)}) - \nabla f(x^{(k)})$, is equivalent to

$$x^{(k+1)} = x^{(k)} - \frac{x^{(k)}}{V(x^{(k)})} \cdot \nabla f(x^{(k)}) = x^{(k)} - D_k^{-1} \nabla f(x^{(k)})$$

where D_k^{-1} is a symmetric positive definite matrix of the form

$$D_k^{-1} = \operatorname{diag}\left(\frac{x_1^{(k)}}{V_1(x^{(k)})}, \dots, \frac{x_n^{(k)}}{V_n(x^{(k)})} \right). \tag{1.55}$$

Therefore it is reasonable to address problem (1.1) by means of a scaled gradient method with steplength equal to 1. In the light of this, the idea proposed in [37] and subsequent works is to adopt the matrix (1.55) into Algorithm 2, with the further request of forcing its eigenvalues to belong to the bounded interval $[1/\mu, \mu]$, in order to comply with STEP 2:

$$(D_k^{-1})_{ii} = \max\left\{\min\left\{\frac{x_i^{(k)}}{V_i(x^{(k)})}, \mu\right\}, \frac{1}{\mu}\right\}, \quad i = 1, \dots, n. \tag{1.56}$$

Note that such a matrix is diagonal, which avoids to introduce significant computational costs in the scheme and, in particular, in the computation of the projection $P_{\Omega,D_k}(\cdot)$.

Steplength choice

Once the scaling matrix is computed, the choice of the steplength α_k has to be considered. Due to the large success of the Barzilai-Borwein rules (1.21) in the context of unconstrained optimization, it is rather natural to extend the various BB-like schemes described in Section 1.1 to the SGP method. A similar extension was first devised for (nonscaled) gradient projection methods in [24], where the authors propose two GP schemes denominated Spectral Projected Gradient (SPG) methods, one performing the linesearch on λ_k along the arc and the other along the feasible direction, which are both equipped with the choice $\alpha_k = \alpha_k^{BB1}$ for the steplength. The theory is extended to scaled gradient projection methods in [25], however the related numerical experience is just concerned with the nonscaled case. In [37] an adaptation of the two BB rules that takes into account the presence of a scaling matrix is devised, by imposing the secant equations (1.19)-(1.20) to the matrix $B(\alpha_k) = (\alpha_k D_k^{-1})^{-1}$, thus obtaining the following rules

$$\alpha_k^{BB1S} = \frac{{s^{(k-1)}}^T D_k D_k s^{(k-1)}}{{s^{(k-1)}}^T D_k y^{(k-1)}} \quad ; \quad \alpha_k^{BB2S} = \frac{{s^{(k-1)}}^T D_k^{-1} y^{(k-1)}}{{y^{(k-1)}}^T D_k^{-1} D_k^{-1} y^{(k-1)}}. \tag{1.57}$$

At this point, inspired by the alternation strategy (1.38) implemented in the framework of nonscaled gradient methods, the authors in [37] propose a steplength updating rule for SGP which adaptively alternates the values provided in (1.57), as detailed in Algorithm 3.

Indeed Algorithm 3 is a modification of rule (1.38) in which the alternation of the two steplengths is determined by means of variable threshold τ_k, instead of the constant parameter τ in (1.38). This trick makes the choice of τ_0 less important for the SGP performance and, in the authors' experience, seems able to avoid the drawbacks due to the use of the same steplength rule in too many consecutive iterations.

In the same spirit, the limited-memory steplength rule devised in [65] and based on the Ritz-like values of the tridiagonal matrix (1.45) can also be exploited in the SGP framework when Ω is the non-negativity constraint set, as suggested in [107]. In the extension of the original scheme to the SGP method, the main change is the definition of a new matrix $\widetilde{G}$ that generalizes the matrix G in (1.40) by taking into account two fundamental elements: the presence of a scaling matrix and the projection onto the feasible set. As concerns the former issue, we recall that applying a scaled gradient method $x^{(k+1)} = x^{(k)} - \alpha_k D_k^{-1} \nabla f(x^{(k)})$, with D_k

Algorithm 3 Steplength Selection (SS) rule

```
IF $k = 0$
    set $\alpha_0 \in [\alpha_{min}, \alpha_{max}]$, $\tau_1 \in (0, 1)$ and a nonnegative integer $M_\alpha$;
ELSE

    IF $s^{(k-1)^T} D_k y^{(k-1)} \leq 0$ THEN
        $\alpha_k^{(1)} = \alpha_{max}$;
    ELSE
        $\alpha_k^{(1)} = \min\left\{\alpha_{max}, \max\{\alpha_{min}, \alpha_k^{BB1S}\}\right\}$;
    ENDIF

    IF $s^{(k-1)^T} D_k^{-1} y^{(k-1)} \leq 0$ THEN
        $\alpha_k^{(2)} = \alpha_{max}$;
    ELSE
        $\alpha_k^{(2)} = \min\left\{\alpha_{max}, \max\{\alpha_{min}, \alpha_k^{BB2S}\}\right\}$;
    ENDIF

    IF $\alpha_k^{(2)} / \alpha_k^{(1)} \leq \tau_k$ THEN
        $\alpha_k = \min\left\{\alpha_j^{(2)},\ j = \max\{1, k - M_\alpha\}, \ldots, k\right\}$;    $\tau_{k+1} = \tau_k \cdot 0.9$;
    ELSE
        $\alpha_k = \alpha_k^{(1)}$;    $\tau_{k+1} = \tau_k \cdot 1.1$.
    ENDIF

ENDIF
```

symmetric and positive definite, to the minimization of a function f is equivalent to performing the change of variables $x = D_k^{-1/2} y$ and addressing the following scaled problem

$$\min_{y \in \mathbb{R}^n} \widetilde{f}(y) \equiv f(D_k^{-1/2} y)$$

by means of a steepest descent method

$$y^{(k+1)} = y^{(k)} - \alpha_k \nabla \widetilde{f}(y^{(k)}) \tag{1.58}$$

with respect to the variable y [22]. The previous remark naturally leads to the idea of applying the limited-memory scheme to the method (1.58) instead of the scaled version of it and, since $\nabla \widetilde{f}(y^{(k)}) = D_k^{-1/2} \nabla f(x^{(k)})$, this suggests to store at each iteration the scaled gradient $D_k^{-1/2} g^{(k)}$ instead of $g^{(k)}$. Concerning the second issue, the non-negativity constraint is addressed by looking at the complementarity condition (1.54) satisfied by the solution of problem (1.1), for which the components of the gradient related to inactive constraints in the solution have to vanish. A way to force the minimization over these components is to store the vectors $\widetilde{g}^{(k)}$ whose j-th entry is given by

$$\widetilde{g}_j^{(k)} = \begin{cases} 0 & \text{if } x_j^{(k)} = 0, \\ \left(\nabla f(x^{(k)})\right)_j & \text{if } x_j^{(k)} > 0. \end{cases} \tag{1.59}$$

The implementation of Fletcher's rule for the constrained case is then based on the storage of the following matrix $\widetilde{G}$

$$\widetilde{G} = \left[D_{k-m}^{-1/2}\widetilde{g}^{(k-m)}, \ldots, D_{k-1}^{-1/2}\widetilde{g}^{(k-1)}\right].$$

The next m Ritz-like values θ_i, $i = 1, \ldots, m$, are then computed by following the same passages included in equations (1.43)-(1.45) with G and $g^{(k)}$ replaced by $\widetilde{G}$ and $D_k^{-1/2}\widetilde{g}^{(k)}$. We remark that, for small m, this generalized limited-memory approach is not much more expensive than any of the BB-like schemes previously described. Indeed, if we assume that D_k is diagonal, each sweep requires

- the computation of m scaled gradients $D_j^{-1/2}\widetilde{g}^{(j)}$ and the $m \times m$ symmetric matrix $\widetilde{G}^T\widetilde{G}$, which can be performed with $m + (m+1)m/2 = (m+3)m/2$ vector-vector products;
- the Cholesky factorization of $\widetilde{G}^T\widetilde{G}$ and the solution of the linear system $R^T r = \widetilde{G}^T D_k^{-1/2}\widetilde{g}^{(k)}$, which are computationally inexpensive if m is a very small number (between 3 and 5).

By contrast, the computation of either the BB1S or BB2S steplengths (1.57) for m iterations needs $3m$ vector-vector products. Therefore, if we assume for example $m = 3$, the limited-memory approach has a computational cost of $\mathcal{O}(9n)$ products exactly as the two BB steplengths.

1.2.3 Computation of the projection

Let us assume that the scaling matrix D_k in the SGP method (1.51) is diagonal, that is $D_k^{-1} = \operatorname{diag}\left(d_1^{(k)}, d_2^{(k)}, \ldots, d_n^{(k)}\right)$. When the feasible set is given by $\Omega = \{x \in \mathbb{R}^n :\ x \geq 0\}$, the projection onto Ω induced by the norm of the matrix D_k is trivial and does not require further investigation. One case of particular interest in this thesis is the following

$$\Omega = \left\{x \in \mathbb{R}^n :\ x \geq 0,\ \sum_{i=1}^n x_i = c\right\} \tag{1.60}$$

where c is a positive constant. When SGP is applied to problem (1.1) subject to (1.60), the projection $P_{\Omega,D_k}(x^{(k)} - \alpha_k D_k^{-1}\nabla f(x^{(k)}))$ must be computed at each iteration which, by relation (1.48), is equivalent to solve the constrained strictly convex quadratic problem

$$\min_{y\in\Omega} \frac{1}{2} y^T D_k y - y^T z \tag{1.61}$$

where $z = D_k(x^{(k)} - \alpha_k D_k^{-1} \nabla f(x^{(k)}))$.
If now we indicate with x^* a solution of (1.1), then the KKT optimality conditions hold, namely there exist Lagrange multipliers $\lambda^* \in \mathbb{R}$ and $\mu^* \in \mathbb{R}^n$ for which

$$\begin{aligned} D_k x^* - z - \lambda^* \boldsymbol{e} - \mu^* &= 0 \\ x^* &\geq 0 \\ \mu^* &\geq 0 \\ {\mu^*}^T x^* &= 0 \\ \sum_{i=1}^{n} x_i^* - c &= 0. \end{aligned}$$

From the first four KKT conditions, we easily obtain x^* and μ^* as functions of λ^*:

$$x_i^*(\lambda^*) = \max\left\{0, d_i^{(k)}(z_i + \lambda^*)\right\}, \quad \mu_i(\lambda^*) = \max\left\{0, -(z_i + \lambda^*)\right\}, \quad i = 1, \ldots, n.$$

Therefore, in order to completely solve the KKT conditions, $x_i^*(\lambda^*)$ must satisfy the fifth KKT condition, i.e. λ^* must be determined as the solution of

$$\sum_{i=1}^{n} x_i^*(\lambda^*) - c = 0.$$

In other words, the computation of the $P_{\Omega, D_k}(x^{(k)} - \alpha_k D_k^{-1} \nabla f(x^{(k)}))$ reduces to the solution of a root-finding problem for a piecewise linear monotonically non-decreasing function, which can be addressed by several linear time algorithms available in the literature (see e.g. [55]).

Chapter 2

Proximal–gradient methods for nondifferentiable optimization

Several applications in signal and image processing are typically reformulated as an optimization problem, in which the objective function is given by the sum of a fit-to-data term, describing the relation between the desired object and the measured data, and possible regularization terms aimed at restricting the search of the object itself to those satisfying specific properties. In other words, one is interested in addressing the following problem

$$\min_{x\in\mathbb{R}^n} f(x) \equiv f_0(x) + f_1(x) \tag{2.1}$$

where, typically, the involved functions satisfy the following properties:

- $f_1 : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ is an extended value function which is proper, convex and lower semicontinuous;
- $f_0 : \mathbb{R}^n \longrightarrow \mathbb{R}$ is a continuously differentiable function on an open set $\Omega_0 \supseteq \mathrm{dom}(f_1)$.

Clearly, formulation (2.1) reduces to the differentiable constrained problem (1.1) of Chapter 1 when the term f_1 is chosen as the indicator function of a non empty, closed and convex subset of $\mathbb{R}^n$, i.e.

$$f_1(x) = \iota_\Omega(x) = \begin{cases} 0, & \text{if } x \in \Omega \\ +\infty, & \text{if } x \notin \Omega. \end{cases}$$

When the function f_1 is nondifferentiable, the optimization techniques analysed in Chapter 1 become inadequate for problem (2.1). Among the several numerical strategies designed to address (2.1), *proximal–gradient methods* [48, 51] have earned a great popularity in the last years for their simplicity and low computational cost per iteration, which make them particularly suited for large-scale optimization problems. Such algorithms deal with the functions f_0 and f_1 *separately*, by alternating a *forward* gradient step on the differentiable (possibly nonconvex) term f_0 with a *backward* proximal step onto the convex (possibly nondifferentiable) term f_1. In particular, the backward step requires the evaluation of the *proximal operator*, which is nothing else than the generalization of the notion of projection onto a convex set to a general convex function. In this light, the proximal–gradient method can be interpreted as the natural extension of the gradient projection method, tailored for problem (1.1), to the more general problem (2.1).

The chapter starts with a self-contained summary of the main notions concerning subdifferential calculus and the proximity operator in Section 2.1, followed by an overview of proximal–gradient methods and related convergence results for the convex case in Section 2.2. Finally, convergence for inexact proximal–gradient methods under the hypothesis that the objective function satisfies the Kurdyka-Łojasiewicz property is discussed in Section 2.3.

2.1 Mathematical background

This section introduces some useful definitions and properties of convex and variational analysis that will be fundamental for the subsequent discussion. A more exhaustive overview of these topics can be found in [130, 131, 143].

2.1.1 Subdifferential calculus

Definition 2.1. *The domain of a function $f : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ is the set* $\mathrm{dom}(f)$ *given by*

$$\mathrm{dom}(f) := \{x \in \mathbb{R}^n : \ f(x) < +\infty\}.$$

Definition 2.2. *A function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is said to be proper if there exists $\bar{x} \in \mathbb{R}^n$ such that $f(\bar{x}) < +\infty$ and $f(x) > -\infty$ for all $x \in \mathbb{R}^n$, namely if* $\mathrm{dom}(f) \neq \emptyset$ *and f is finite on* $\mathrm{dom}(f)$.

Definition 2.3. *The epigraph of a function $f : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ is the set* $\mathrm{epi}(f)$ *given by*

$$\mathrm{epi}(f) := \{(x,t) \in \mathbb{R}^n \times \mathbb{R} : f(x) \leq t\}.$$

Definition 2.4. *[131, Definition 1.5] A function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is lower semicontinuous (lsc) at x if*

$$f(x) = \liminf_{y \to x} f(y) = \sup_{\delta > 0} \left(\inf_{y \in B(x,\delta)} f(y) \right). \tag{2.2}$$

Similarly, f is upper semicontinuous at x if

$$f(x) = \limsup_{y \to x} f(x) = \inf_{\delta > 0} \left(\sup_{y \in B(x,\delta)} f(y) \right). \tag{2.3}$$

Remark 2.1. The function f is continuous at x if and only if f is both lower and upper semicontinuous at x.

Proposition 2.1. *Consider a function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$. The following conditions are equivalent:*

(i) f is lower semicontinuous on $\mathbb{R}^n$;

(ii) the epigraph $\mathrm{epi}(f)$ *is a closed subset of $\mathbb{R}^n \times \mathbb{R}$;*

(iii) for every $\alpha \in \bar{\mathbb{R}}$, the level set $[f \leq \alpha]$ is a closed subset of $\mathbb{R}^n$.

Proof. Before proving the equivalence of items (i), (ii) and (iii), we recall that the lower limit of a function can be characterized in the following way [131, Lemma 1.7]

$$\liminf_{y \to x} f(y) = \min\{\alpha \in \bar{\mathbb{R}} : \exists \, \{x^{(k)}\}_{k\in\mathbb{N}} \subseteq \mathbb{R}^n \text{ such that } x^{(k)} \to x, \; f(x^{(k)}) \to \alpha\}. \tag{2.4}$$

(i)⇒(ii). Suppose $(x^{(k)}, \alpha^{(k)}) \in \mathrm{epi}(f)$ and $(x^{(k)}, \alpha^{(k)}) \to (x, \alpha)$ with $\alpha \in \mathbb{R}$. We have $x^{(k)} \to x$, $\alpha^{(k)} \to \alpha$ with $\alpha^{(k)} \geq f(x^{(k)})$ and must prove that $\alpha \geq f(x)$, so that $(x, \alpha) \in \mathrm{epi}(f)$. Let $\beta \in \bar{\mathbb{R}}$ be a limit point of the sequence $\{f(x^{(k)})\}_{k\in\mathbb{N}}$, then there is $\{k_j\}_{j\in\mathbb{N}} \subseteq \mathbb{N}$ such that $f(x^{k_j}) \to \beta$. Since $\alpha^{(k)} \geq f(x^{(k)})$ for all k, it follows that $\alpha \geq \beta$ and, because of (2.4), it is also $\beta \geq \liminf_{y\to x} f(y)$. Then the lower semicontinuity of f allows to conclude that $\alpha \geq f(x)$.

(ii)⇒(iii). If $\mathrm{epi}(f)$ is a closed subset of $\mathbb{R}^n \times \mathbb{R}$, then for any $\alpha \in \mathbb{R}$ the set

$$\mathrm{epi}(f) \cap (\mathbb{R}^n \times \{\alpha\}) = \{(x, \alpha) \in \mathbb{R}^n \times \mathbb{R} : \; f(x) \leq \alpha\} = [f \leq \alpha] \times \{\alpha\}$$

is closed as well. Therefore, the level set $[f \leq \alpha]$ must be closed. If $\alpha = -\infty$, the corresponding level set is $[f \leq -\infty] = [f = -\infty] = \cap_{\alpha\in\mathbb{R}}[f \leq \alpha]$, namely an intersection of closed sets and thus itself closed. Finally, if $\alpha = +\infty$, the level set is the whole space $\mathbb{R}^n$.

(iii)⇒(i) In order to establish that f is lsc at any point $x \in \mathbb{R}^n$, it suffices to prove that $\bar{\alpha} = \liminf_{y\to x} f(y) \geq f(x)$, since the opposite inequality is always satisfied. Since the case $\bar{\alpha} = \infty$ is trivial, suppose $\bar{\alpha} < \infty$. Relation (2.4) guarantees the existence of a sequence $\{x^{(k)}\}_{k\in\mathbb{N}} \subseteq \mathbb{R}^n$ such that $x^{(k)} \to x$ and $f(x^{(k)}) \to \bar{\alpha}$. Hence, for any $\alpha > \bar{\alpha}$ and for all sufficiently large k, it will be true that $f(x^{(k)}) \leq \alpha$ or, equivalently, that $x^{(k)} \in [f \leq \alpha]$. Since $x^{(k)} \to x$ and $[f \leq \alpha]$ is closed by assumption, it follows that $x \in [f \leq \alpha]$ for all $\alpha > \bar{\alpha}$, that is $f(x) \leq \alpha$ for all $\alpha > \bar{\alpha}$. Then this implies $f(x) \leq \bar{\alpha}$. □

Proposition 2.2. *Let $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ be a proper function. The following facts are equivalent:*

(i) f is convex, i.e.

$$f((1-\lambda)x + \lambda y) \leq (1-\lambda)f(x) + \lambda f(y), \quad \forall \; x, y \in \mathbb{R}^n, \; \forall \; \lambda \in (0,1).$$

(ii) (Jensen's inequality) For any $x_1, \ldots, x_n \in \mathrm{dom}(f)$ and $\lambda_1, \ldots, \lambda_n \in \mathbb{R}$ such that $\lambda_i \geq 0$, $\sum_{i=1}^n \lambda_i = 1$, we have

$$f\left(\sum_{i=1}^n \lambda_i x_i\right) \leq \sum_{i=1}^n \lambda_i f(x_i).$$

(iii) The epigraph epi(f) *is a convex subset of* $\mathbb{R}^n \times \mathbb{R}$.

Proof. See [131, Theorem 2.2, Proposition 2.4]. □

Definition 2.5. *The conjugate function* $f^* : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ *of a convex function* $f : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ *is defined as*

$$f^*(y) = \sup_{x \in \mathbb{R}^n} y^T x - f(x).$$

The biconjugate function $f^{**} : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ *of* f *is defined as* $f^{**} := (f^*)^*$, *i.e.*

$$f^{**}(y) = \sup_{x \in \mathbb{R}^n} y^T x - f^*(x).$$

Example 2.1. The conjugate of the *indicator function* ι_Ω of a non empty set $\Omega \subseteq \mathbb{R}^n$ is

$$\iota_\Omega^*(y) = \sup_{x \in \Omega} y^T x, \quad \forall\, y \in \mathbb{R}^n$$

namely the *support function* of Ω. In particular:

- if Ω is the nonnegative orthant, then $\iota^*_{\mathbb{R}^n_{\geq 0}} = \iota_{\mathbb{R}^n_{\leq 0}}$;
- if Ω is a linear subspace, then $\iota^*_\Omega = \iota_{\Omega^\perp}$.

Example 2.2. Consider $f(x) = \lambda \|x\|$ where $\lambda \in \mathbb{R}_{>0}$. Then

$$\begin{aligned} f^*(y) &= \sup_{x \in \mathbb{R}^n} y^T x - \lambda \|x\| \\ &= \sup_{t \in \mathbb{R}_{\geq 0}} \left(\sup_{\|x\|=1} y^T(tx) - t\lambda\|x\| \right) \\ &= \sup_{t \in \mathbb{R}_{\geq 0}} t(\|y\| - \lambda) \end{aligned}$$

where the last equality is obtained by recalling that $\|y\| = \sup_{\|x\|=1} y^T x$. Therefore

$$f^*(y) = \begin{cases} 0, & \text{if } \|y\| \leq \lambda \\ \infty, & \text{otherwise} \end{cases} = \iota_{B(0,\lambda)}(y).$$

Example 2.3. Let $f(x) = \frac{1}{2} x^T A x + b^T x$, where $A \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix and $b \in \mathbb{R}^n$. Then the conjugate function of f is

$$f^*(y) = \sup_{x \in \mathbb{R}^n} y^T x - f(x) = \sup_{x \in \mathbb{R}^n} \left[-\frac{1}{2} x^T A x + (y-b)^T x \right] \equiv \varphi(x).$$

Since φ is concave and differentiable, its maximum is attained in the unique point $x^* \in \mathbb{R}^n$ such that $\nabla \varphi(x^*) = 0$, that is $x^* = A^{-1}(y - b)$. Then

$$f^*(y) = \varphi(x^*) = \frac{1}{2}(y-b)^T A^{-1}(y-b). \tag{2.5}$$

Proposition 2.3. *Suppose that $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is given by a separable sum of convex functions, i.e.*

$$f(x) = \sum_{i=1}^{r} f_i(x_i),$$

where $f_i : \mathbb{R}^{n_i} \to \bar{\mathbb{R}}$ is convex for $i = 1, \dots, r$ and $\sum_{i=1}^{r} n_i = n$. Then

$$f^*(y) = \sum_{i=1}^{r} f_i^*(y_i), \quad \forall\, y \in \mathbb{R}^n.$$

Proof. From the definition of conjugate function we have

$$\begin{aligned} f^*(y) &= \sup_{x \in \mathbb{R}^n} \left(y^T x - \sum_{i=1}^{r} f_i(x_i) \right) = \sup_{x \in \mathbb{R}^n} \left(\sum_{i=1}^{r} y_i^T x_i - f_i(x_i) \right) \\ &= \sum_{i=1}^{r} \left(\sup_{x_i \in \mathbb{R}^n} y_i^T x_i - f_i(x_i) \right) = \sum_{i=1}^{r} f_i^*(y_i). \end{aligned}$$

□

Lemma 2.1. *Let $f : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ be a convex function and $f^* : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ its conjugate function. Then the following inequalities hold true:*

(i) (Fenchel's inequality) $f^*(y) + f(x) \geq y^T x, \quad \forall x, y \in \mathbb{R}^n$.

(ii) $f(x) \geq f^{**}(x), \quad \forall\, x \in \mathbb{R}^n$.

Proof. (i) It is an immediate consequence of the definition of conjugate function.
(ii) The Fenchel's inequality and the definition of biconjugate function lead to the following relations:

$$\begin{aligned} f(x) \geq y^T x - f^*(y)\ \forall x, y \in \mathbb{R}^n \quad &\Longleftrightarrow \quad f(x) \geq \sup_{y \in \mathbb{R}^n} y^T x - f^*(y)\ \forall x \in \mathbb{R}^n \\ &\Longleftrightarrow \quad f(x) \geq f^{**}(x)\ \forall x \in \mathbb{R}^n. \end{aligned}$$

□

Theorem 2.1 (Biconjugate theorem)**.** *If $f : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ is a lower semicontinuous and convex function then $f^{**} = f$.*

Proof. In the light of item (ii) of Lemma 2.1, it suffices to prove that $f(x) \leq f^{**}(x)$ for all $x \in \mathbb{R}^n$. Let us suppose by contradiction that there exists $x \in \mathbb{R}^n$ such that $f(x) > f^{**}(x)$ or, equivalently, $(x, f^{**}(x)) \notin \mathrm{epi}(f)$. Since f is lsc and convex on $\mathbb{R}^n$ and in virtue of Proposition 2.1 and 2.2, epi(f) is closed and convex and consequently, by the hyperplane separation theorem [131, Theorem 2.39], it is possible to find a strict separating hyperplane verifying the following inequality:

$$(a, b)^T (z - x, s - f^{**}(x)) \leq c < 0, \qquad \forall\, (z, s) \in \mathrm{epi}(f) \tag{2.6}$$

for some $a \in \mathbb{R}^n, b, c \in \mathbb{R}$. We observe that b must be nonpositive, since $(z, s+n) \in \operatorname{epi}(f)$ for every $n \in \mathbb{N}$ and $b > 0$ gives a contradiction as $n \to +\infty$. Thus we have two possible cases:

(i) if $b < 0$, we define $y = -a/b$ and, if we divide by $-b$ and maximize the left hand-side of (2.6) over $(z, s) \in \operatorname{epi}(f)$, we obtain:

$$f^*(y) - y^T x + f^{**}(x) \leq \frac{-c}{b} < 0.$$

This is in contrast with the Fenchel's inequality;

(ii) if $b = 0$, we let $\hat{y} \in \operatorname{dom} f^*$ and add a sufficient small multiple of $(\hat{y}, -1)$ to (a, b) thus obtaining

$$(a + \epsilon\hat{y}, -\epsilon)^T (z - x, s - f^{**}(x)) \leq c + \epsilon(f^*(\hat{y}) - \hat{y}^T x + f^{**}(x)) < 0, \qquad \forall\ (z, s) \in \operatorname{epi}(f).$$

By applying the same argument used for $b < 0$, the contradiction is reached.

□

Definition 2.6. *[131, Definition 8.3] Let $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ and $x \in \operatorname{dom}(f)$. The* Fréchet subdifferential *of f at x is the set*

$$\hat{\partial} f(x) = \left\{ v \in \mathbb{R}^n : \liminf_{y \to x, y \neq x} \frac{1}{\|x - y\|} (f(y) - f(x) - (y - x)^T v) \geq 0 \right\}.$$

*The limiting-*subdifferential *(or simply subdifferential) of f at x is defined as*

$$\begin{aligned} \partial f(x) = \ \ & \{v \in \mathbb{R}^n : \ \exists\ \{y^{(k)}\}_{k \in \mathbb{N}} \subseteq \mathbb{R}^n,\ v^{(k)} \in \hat{\partial} f(y^{(k)})\ \forall k \in \mathbb{N} \text{ such that} \\ & \qquad\qquad y^{(k)} \to x,\ f(y^{(k)}) \to f(x) \text{ and } v^{(k)} \to v\}. \end{aligned}$$

Finally, we define $\operatorname{dom}(\partial f) = \{x \in \operatorname{dom}(f) : \partial f(x) \neq \emptyset\}$.

Remark 2.2. The above definition implies that $\hat{\partial} f(x) \subseteq \partial f(x)$ for all $x \in \mathbb{R}^n$, where the first set is convex and closed while the second one is closed [131, Theorem 8.6].

Lemma 2.2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be as in problem* (2.1)*, where f_0 is a continuously differentiable function on an open set $\Omega_0 \supseteq \operatorname{dom}(f_1)$. Then:*

(i) $\hat{\partial} f_0(x) = \partial f_0(x) = \{\nabla f_0(x)\}, \quad \forall\ x \in \Omega_0$*;*

(ii) $\partial f(x) = \{\nabla f_0(x)\} + \partial f_1(x), \quad \forall\ x \in \operatorname{dom}(f_1)$.

Proof. See [131, Exercise 8.8]. □

Lemma 2.3. *Let $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ be a proper, convex function. Then for any $x \in \operatorname{dom}(f)$*

$$\hat{\partial} f(x) = \partial f(x) = \{v \in \mathbb{R}^n : \ f(y) \geq f(x) + (y - x)^T v \ \ \forall y \in \mathbb{R}^n\}. \tag{2.7}$$

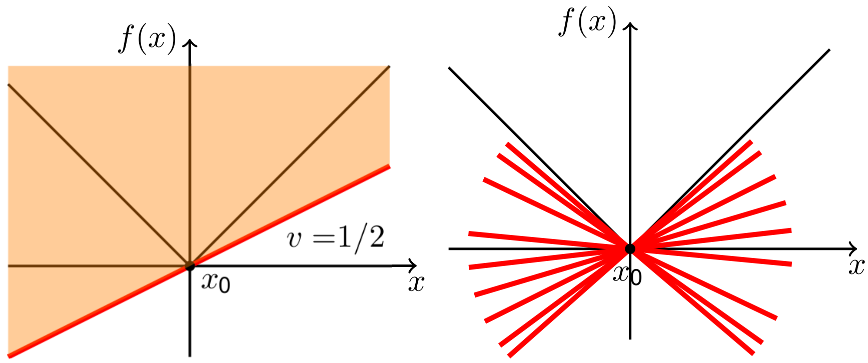


Figure 2.1: Subgradients of the function $f(x) = |x|$ at $x_0 = 0$. Left: the value $v = 1/2$ identifies a supporting line at $(0,0)$ and thus $v \in \partial f(0)$. Right: graphical representation of the set $\partial f(0) = [-1,1]$.

Proof. See [131, Proposition 8.12]. □

Remark 2.3. Lemma 2.3 asserts that, when the function is convex, both the Fréchet and limiting-subdifferential of Definition 2.6 coincides with the usual subdifferential of convex analysis [130, p. 214], also known as *Fenchel subdifferential.* In this special case, the set $\partial f(x)$ has a simple geometric interpretation, which is the following: $v \in \partial f(x)$ if and only if the graph of the affine function $h(y) = f(x) + (y-x)^T v$ is a non-vertical supporting hyperplane to the convex set epi(f) at the point $(x, f(x))$, as depicted in Figure 2.1.

Example 2.4. Let $f(x) = |x|$. By using item (i) of Lemma 2.2 and Lemma 2.3, it is easy to see that

$$\partial f(x) = \begin{cases} [-1,1], & \text{if } x = 0 \\ \{x/|x|\}, & \text{if } x \neq 0. \end{cases}$$

Note that the subgradient of f is an interval at the origin (see Figure 2.1).

Example 2.5. Consider the *indicator function* ι_Ω of a non empty, convex set $\Omega \subseteq \mathbb{R}^n$. By directly using equation (2.7), we have

$$\partial \iota_\Omega(x) = \{v \in \mathbb{R}^n : \ v^T(y-x) \leq 0\} = N_\Omega(x),$$

where $N_\Omega(x)$ denotes the *normal cone* to the convex set Ω at the point $x \in \Omega$ [130, p. 15].

We now define the so-called $\epsilon-$subdifferential, which represents an approximation of the subdifferential (2.7) for convex functions.

Definition 2.7. *[143, p. 82] Let $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ be a proper, convex function and $\epsilon \in \mathbb{R}_{\geq 0}$. The ϵ-subdifferential of f at $x \in \mathrm{dom}(f)$ is the set*

$$\partial_\epsilon f(x) = \{v \in \mathbb{R}^n : f(y) \geq f(x) + (y-x)^T v - \epsilon \quad \forall y \in \mathbb{R}^n\}. \tag{2.8}$$

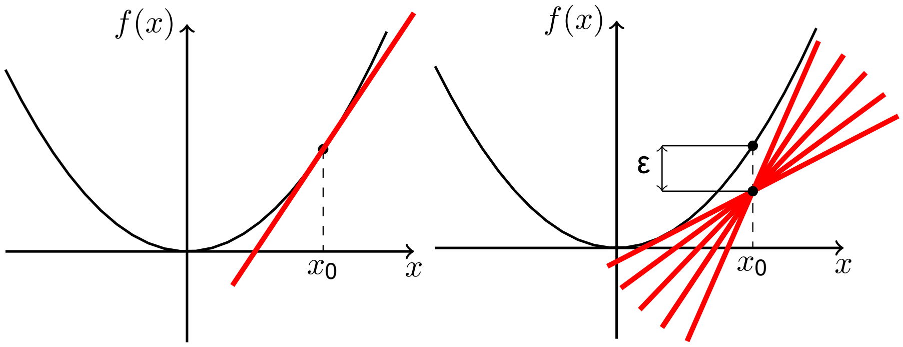


Figure 2.2: Exact and approximate subdifferential of the function $f(x) = x^2/2$ at the point x_0. Left: the set $\partial f(x_0) = \{x_0\}$ contains a unique point. Right: the set $\partial_\epsilon f(x_0) = [x_0 - \sqrt{2\epsilon}, x_0 + \sqrt{2\epsilon}]$ is an interval.

Remark 2.4. (i) If $x \notin \mathrm{dom}(f)$, then $\partial_\epsilon f(x) := \emptyset$ for any $\epsilon \in \mathbb{R}_{\geq 0}$.
(ii) If $\epsilon = 0$, then $\partial_\epsilon f(x) = \partial f(x)$ (see Lemma 2.3).
(iii) If $0 \leq \epsilon_1 \leq \epsilon_2$, then $\partial_{\epsilon_1} f(x) \subseteq \partial_{\epsilon_2} f(x)$, $\forall\, x \in \mathrm{dom}(f)$.

The previous remark leads us to ask whether, for a given point $x \in \mathrm{dom}(f)$, the $\epsilon-$subdifferential $\partial_\epsilon f(x)$ is a greater set than the exact subdifferential $\partial f(x)$. Indeed this is often the case, as suggested by the following elementary examples.

Example 2.6. Let $f(x) = x^2/2$. Since f is continuously differentiable on $\mathbb{R}$, by item (i) of Lemma 2.2 we have $\partial f(x) = \{\nabla f(x)\} = \{x\}$. However, it is easy to show that

$$\partial_\epsilon f(x) = \left[x - \sqrt{2\epsilon}, x + \sqrt{2\epsilon}\right] = \partial f(x) + \left[\sqrt{2\epsilon}, \sqrt{2\epsilon}\right].$$

A graphical representation of both sets is provided in Figure 2.2.

Example 2.7. Let $f(x) = |x|$. Then

$$\partial_\epsilon f(x) = \begin{cases} [-1, -1 - \epsilon/x], & \text{if } x < -\epsilon/2, \\ [-1, 1], & \text{if } -\epsilon/2 \leq x \leq \epsilon/2, \\ [1 - \epsilon/x, 1], & \text{if } x > \epsilon/2. \end{cases}$$

In this case, $\partial_\epsilon f(x) = \partial f(x) \Longleftrightarrow x = 0$.

Proposition 2.4. *Let $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ be a proper, convex, lsc function and $\epsilon \in \mathbb{R}_{\geq 0}$. Then we have:*

$$v \in \partial_\epsilon f(x) \Longleftrightarrow x \in \partial_\epsilon f^*(v). \tag{2.9}$$

Proof. We observe that

$$\begin{aligned}
v \in \partial_\epsilon f(x) &\iff (y-x)^T v \leq f(y) - f(x) + \epsilon \quad \forall y \in \mathbb{R}^n \\
&\iff y^T v - f(y) \leq x^T v - f(x) + \epsilon \quad \forall y \in \mathbb{R}^n \\
&\iff f^*(v) \leq x^T v - f(x) + \epsilon \\
&\iff f(x) + f^*(v) \leq x^T v + \epsilon. \qquad (2.10)
\end{aligned}$$

Since f is lsc, Theorem 2.1 holds and thus it is true that

$$f^{**}(x) + f^*(v) = f(x) + f^*(v).$$

The thesis follows by combining the relation above with (2.10). □

Example 2.8. Let $f(x) = \frac{1}{2}x^T Ax + b^T x$ as in Example 2.3. Equivalence (2.10) can be rewritten as

$$v \in \partial_\epsilon f(x) \iff \frac{1}{2}(v-b)^T A^{-1}(v-b) + f(x) \leq v^T x + \epsilon.$$

By setting $v = Ax + b + e$, with $e \in \mathbb{R}^n$, in the above equation and applying some algebra, we obtain

$$\partial_\epsilon f(x) = \{\nabla f(x)\} + \left\{ e \in \mathbb{R}^n : \ \frac{\|e\|^2_{A^{-1}}}{2} \leq \epsilon \right\} = \left\{ Ax + b + e : \ \frac{\|e\|^2_{A^{-1}}}{2} \leq \epsilon \right\}. \qquad (2.11)$$

Proposition 2.5. *Let $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ be a proper, convex function. Then*

$$f \textit{ is lower semicontinuous at } x \in \mathrm{dom}(f) \iff \partial_\epsilon f(x) \neq \emptyset, \quad \forall \ \epsilon \in \mathbb{R}_{>0}.$$

Proof. See [143, Theorem 2.4.4]. □

Proposition 2.6. *[143, Theorem 2.4.2(ix)] Let $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ be a proper, convex, lsc function, $\{\epsilon_k\}_{k\in\mathbb{N}} \subseteq \mathbb{R}_{\geq 0}$, $\epsilon \in \mathbb{R}_{\geq 0}$, and $\{(x^{(k)}, v^{(k)})\}_{k\in\mathbb{N}} \subseteq \mathbb{R}^n \times \mathbb{R}^n$ such that*

$$(x^{(k)}, v^{(k)}) \in \text{graph } \partial_{\epsilon_k} f = \{(x, x^*) \in \mathbb{R}^n \times \mathbb{R}^n : x^* \in \partial_{\epsilon_k} f(x)\}.$$

If $(x^{(k)}, v^{(k)}) \to (x, v)$ and $\epsilon_k \to \epsilon$ as $k \to +\infty$, then $(x, v) \in \text{graph } \partial_\epsilon f$.

Proof. Since $v^{(k)} \in \partial_{\epsilon_k} f(x^{(k)})$, by Definition 2.7 we have

$$f(y) \geq f(x^{(k)}) + (y - x^{(k)})^T v^{(k)} - \epsilon_k, \qquad \forall \ y \in \mathbb{R}^n.$$

The thesis follows by taking the lower limit over k and from the lower semicontinuity of f. □

Proposition 2.7. *Let $f, g : \mathbb{R}^n \to \bar{\mathbb{R}}$, $h : \mathbb{R}^m \to \bar{\mathbb{R}}$ be proper, convex, lower semicontinuous functions and $\epsilon \in \mathbb{R}_{\geq 0}$. Then, we have the following properties:*

(i) if $g(x) = f(x) + c$ with $c \in \mathbb{R}$, then $\partial_\epsilon g(x) = \partial_\epsilon f(x)$, $\forall \ x \in \mathrm{dom}(f)$;

(ii) *if* $g(x) = f(x) + c^T x$ *with* $c \in \mathbb{R}$, *then* $\partial_\epsilon g(x) = \partial_\epsilon f(x) + \{c\}$, $\forall\ x \in \mathrm{dom}(f)$;

(iii) *if* $g(x) = \alpha f(x)$ *with* $\alpha \in \mathbb{R}_{>0}$, *then* $\partial_\epsilon g(x) = \alpha \partial_{\epsilon/\alpha} f(x)$, $\forall\ x \in \mathrm{dom}(f)$;

(iv) *if* $g(x) = f(\alpha x)$ *with* $\alpha \in \mathbb{R} \setminus \{0\}$, *then* $\partial_\epsilon g(x) = \alpha \partial_\epsilon f(\alpha x)$, $\forall\ x \in \mathrm{dom}(f)$;

(v) *if* $h(x) = f(Ax)$ *with* $A \in \mathbb{R}^{n\times m}$ *and* $\mathrm{Im}(A) \cap \mathrm{int}(\mathrm{dom}(f)) \neq \emptyset$, *then*

$$\partial_\epsilon h(x) = A^T \partial_\epsilon f(Ax), \quad \forall\ x \in \mathbb{R}^m :\ Ax \in \mathrm{dom}(f);$$

(vi) *if* $\mathrm{ri}(\mathrm{dom}(f)) \cap \mathrm{ri}(\mathrm{dom}(g)) \neq \emptyset$, *then*

$$\partial_\epsilon (f+g)(x) = \bigcup_{0 \le \epsilon_1 + \epsilon_2 \le \epsilon} \partial_{\epsilon_1} f(x) + \partial_{\epsilon_2} g(x), \quad \forall\ x \in \mathrm{dom}(f) \cap \mathrm{dom}(g);$$

(vii) *if* $\mathrm{ri}(\mathrm{dom}(f)) \cap \mathrm{ri}(\mathrm{dom}(g)) \neq \emptyset$, *where* $\mathrm{ri}(\cdot)$ *indicates the relative interior of a set, and* $\alpha_1, \alpha_2 \ge 0$, *then*

$$\partial(\alpha_1 f + \alpha_2 g)(x) = \alpha_1 \partial f(x) + \alpha_2 \partial g(x), \quad \forall\ x \in \mathrm{dom}(f) \cap \mathrm{dom}(g).$$

(viii) *If* $f(x) = \sum_{i=1}^r f_i(x_i)$, *with* $f_i : \mathbb{R}^{n_i} \to \bar{\mathbb{R}}$ *proper, convex for* $i = 1, \dots, r$ *and* $\sum_{i=1}^r n_i = n$, *then*

$$\partial f(x) = \prod_{i=1}^r \partial f_i(x_i) = \left(\partial f_1(x_1), \dots, \partial f_r(x_r)\right), \quad \forall\ x \in \mathrm{dom}(f).$$

Proof. Items (i)-(iv) follow by directly applying Definition 2.7 of $\epsilon-$subdifferential. The proof of item (v) can be found in [84, Theorem 3.2.1] and the one of item (vi) in [84, Theorem 3.1.1]. Item (vii) is obtained by combining items (vi) and (iii). Finally, item (viii) is proved in [143, Corollary 2.4.5] □

2.1.2 Optimality conditions

In the nondifferentiable case, it is possible to formulate the necessary optimality condition for a point to be a minimum of a function f in terms of its subdifferential. The following result is the analogous of Theorem 1.1 and 1.2 given in the differentiable case.

Proposition 2.8. *Let* $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ *be a proper function.*

(i) *If* $x \in \mathbb{R}^n$ *is a local minimizer of* f, *then* $0 \in \partial f(x)$.

(ii) *If* f *is also convex,* $x \in \mathbb{R}^n$ *is a global minimizer if and only if* $0 \in \partial f(x)$.

Proof. (i) If x is a local minimizer, then there exists $\rho > 0$ such that $f(y) \ge f(x)$ for all $y \in B(x, \rho)$, which implies that $0 \in \hat{\partial} f(x)$. Remark 2.2 allows to conclude the proof.
(ii) The implication from left to right follows from item (i), while the converse is obtained by substituting $v = 0$ in Lemma 2.3. □

Definition 2.8. *A point $x \in \mathbb{R}^n$ is* stationary *for a function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ if $x \in \mathrm{dom}(f)$ and $0 \in \partial f(x)$.*

Definition 2.8 may be equivalently reformulated in terms of the directional derivative of f.

Definition 2.9. *[130, p.213] Let $f : \mathbb{R}^n \to \bar{\mathbb{R}}$. The* one sided directional derivative *of f at x with respect to a vector $d \in \mathbb{R}^n$ is defined as*

$$f'(x; d) = \lim_{\lambda \downarrow 0} \frac{f(x + \lambda d) - f(x)}{\lambda} \tag{2.12}$$

if the limit on the right-hand side exists in $\bar{\mathbb{R}}$.

Remark 2.5. (i) When f is continuously differentiable in a neighbourhood of x, $f'(x; d) = \nabla f(x)^T d$.
(ii) When f is convex and $x \in \mathrm{dom}(f)$, the limit at the right-hand side of (2.12) exists for any $d \in \mathbb{R}^n$ and $f'(x; d) = \inf_{\lambda > 0}(f(x + \lambda d) - f(x))/\lambda$ [130, Theorem 23.1]. As a consequence, $f(x; d) \leq f(x + d) - f(x)$.

Proposition 2.9. *Let $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ be as in problem* (2.1)*, where f_0 is a continuously differentiable function on an open set $\Omega_0 \supseteq \mathrm{dom}(f_1)$ and f_1 is convex. Then*

$$x \in \mathrm{dom}(f) \text{ is a stationary point for } f \iff f'(x; d) \geq 0, \ \forall \ d \in \mathbb{R}^n.$$

Proof. From Remark 2.5 and the linearity of limit, the function f admits directional derivative on its domain and

$$f'(x, d) = f_0'(x, d) + f_1'(x, d), \quad \forall \ x \in \mathrm{dom}(f), \ \forall \ d \in \mathbb{R}^n.$$

The following relations hold:

$$\begin{aligned}
0 \in \partial f(x) &\iff 0 \in \{\nabla f_0(x)\} + \partial f_1(x) \quad \text{(Lemma 2.2)} \\
&\iff -\nabla f_0(x) \in \partial f_1(x) \\
&\iff f_1(y) \geq f_1(x) - \nabla f_0(x)^T (y - x), \quad \forall \ y \in \mathbb{R}^n \\
&\iff f_1(x + \lambda d) \geq f_1(x) - \lambda f_0'(x; d), \quad \forall \ d \in \mathbb{R}^n, \ \forall \ \lambda \in (0, 1) \\
&\iff \frac{f_1(x + \lambda d) - f_1(x)}{\lambda} + f_0'(x; d) \geq 0, \quad \forall \ d \in \mathbb{R}^n, \ \forall \ \lambda \in (0, 1) \\
&\iff f'(x; d) \geq 0, \forall \ d \in \mathbb{R}^n
\end{aligned}$$

where the last equivalence is obtained by taking the infimum on the left-hand side of the inequality. □

Definition 2.10. *A vector $d \in \mathbb{R}^n$ is a* descent direction *for f at $x \in \mathrm{dom}(f)$ if $f'(x; d) < 0$.*

In the light of Proposition 2.9, when $f_1 = \iota_\Omega$ with $\Omega \subseteq \mathbb{R}^n$ non empty, closed and convex set, Definition 2.8 coincides with the stationarity condition (1.46) given in the differentiable case and, in the same setting, we have $f'(x; d) = \nabla f(x)^T d$ (see item (i) of Remark 2.5), hence also the classical definition of descent direction is recovered.

2.1.3 The proximal operator

The notion of proximal (or proximity) operator was first introduced by Moreau in [99]. Here we give its most general definition with respect to a symmetric positive definite matrix.

Definition 2.11. *[66, §2.3] The* proximity *operator associated to a function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ in the metric induced by a symmetric positive definite matrix D is defined as*

$$\operatorname{prox}_f^D(x) = \arg\min_{z\in\mathbb{R}^n} f(z) + \frac{1}{2}\|z-x\|_D^2, \quad \forall x \in \mathbb{R}^n. \tag{2.13}$$

Remark 2.6. When $D = I_n$, we write $\operatorname{prox}_f^{I_n} = \operatorname{prox}_f$.

Note that, in general, $\operatorname{prox}_f^D : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is a multi-valued map, and it might also happen that $\operatorname{prox}_f^D(x) = \emptyset$ at some point x. However, existence and uniqueness of the proximal point may be guaranteed under convexity and lower semicontinuity assumptions.

Proposition 2.10. *If $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is proper, convex and lower semicontinuous, then* $\operatorname{prox}_f^D(x)$ *exists and is unique for all $x \in \mathbb{R}^n$ and*

$$y = \operatorname{prox}_f^D(x) \iff D(x-y) \in \partial f(y). \tag{2.14}$$

Proof. The function $\varphi(z) = f(z) + \frac{1}{2}\|z-x\|_D^2$ is strictly convex and, thus, it admits at most one minimum point. Furthermore, since φ is also strongly convex, it is coercive and therefore the minimum point exists and is unique. By applying the first order optimality condition to the convex function φ we have

$$\begin{aligned} y = \operatorname{prox}_f^D(x) &\iff 0 \in \partial\varphi(y) && \text{(item (ii) of Proposition 2.8)}\\ &\iff 0 \in \partial f(y) + D(y-x) && \text{(item (vii) of Proposition 2.7)}\\ &\iff D(x-y) \in \partial f(y). \end{aligned}$$

□

Remark 2.7. By setting $w = D(x-y)$ in equation (2.14), it follows that $w \in \partial f(y)$ if and only if $y = \operatorname{prox}_f^D(y + D^{-1}w)$.

Example 2.9. The proximal operator of the indicator function ι_Ω with $\Omega \subseteq \mathbb{R}^n$ non empty, closed and convex set, coincides with the projection operator (1.47):

$$\operatorname{prox}_{\iota_\Omega}^D(x) = P_{\Omega,D}(x) = \operatorname*{argmin}_{z\in\Omega} \|z-x\|_D^2.$$

Proximity operators are therefore a generalization of projection operators.

The proximal operator allows to give a further equivalent definition of stationary point for problem (2.1), in analogy with what already seen for the differentiable case in Lemma 1.1.

Proposition 2.11. *Let $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ be as in problem* (2.1)*, where f_0 is a continuously differentiable function on an open set $\Omega_0 \supseteq \mathrm{dom}(f_1)$ and f_1 is proper, convex and lower semicontinuous. Fix $\alpha \in \mathbb{R}_{>0}$ and D symmetric positive definite matrix. Then*

$$x^* \text{ is stationary for } f \iff x^* = \mathrm{prox}^D_{\alpha f_1}(x^* - \alpha D^{-1} \nabla f_0(x^*)).$$

Proof. By item (ii) of Lemma 2.2, we have $\partial f(x^*) = \{\nabla f_0(x^*)\} + \partial f_1(x^*)$. Therefore, the following equivalences hold:

$$\begin{aligned} 0 \in \partial f(x^*) &\iff 0 \in \alpha \left(\{\nabla f_0(x^*)\} + \partial f_1(x^*)\right) \\ &\iff -\alpha \nabla f_0(x^*) \in \partial(\alpha f_1)(x^*). \end{aligned}$$

The thesis now follows by recalling Remark 2.7. □

Definition 2.12. *Let $f : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ be a proper, convex function. The* resolvent *of the subdifferential ∂f with respect to the symmetric positive definite matrix D is the mapping $(I_n + D^{-1}\partial f)^{-1} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ defined as*

$$(I_n + D^{-1}\partial f)^{-1}(x) = \left\{ y \in \mathbb{R}^n : \ x \in (I_n + D^{-1}\partial f)(y) \right\}, \quad \forall \ x \in \mathbb{R}^n.$$

Proposition 2.12. *Let $f : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ be a proper, convex and lowersemicontinuous function and D a symmetric positive definite matrix. Then*

$$(I_n + D^{-1}\partial f)^{-1}(x) = \mathrm{prox}^D_f(x), \quad \forall \ x \in \mathbb{R}^n$$

and thus $(I_n + D^{-1}\partial f)^{-1}$ is single-valued.

Proof. By Definition 2.12 of resolvent, we have

$$\begin{aligned} y \in (I_n + D^{-1}\partial f)^{-1}(x) &\iff x \in (I_n + D^{-1}\partial f)(y) = y + D^{-1}\partial f(y) \\ &\iff (x - y) \in D^{-1}\partial f(y) \\ &\iff D(x - y) \in \partial f(y) \\ &\iff y = \mathrm{prox}^D_f(x) \end{aligned}$$

where the last equivalence follows from (2.14). □

Lemma 2.4. *Let $f : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ be a proper, convex and lower semicontinuous function and $D \in \mathcal{M}_\mu$. Then the proximal operator prox^D_f is Lipschitz continuous with constant μ^2, i.e.*

$$\| \mathrm{prox}^D_f(x) - \mathrm{prox}^D_f(\tilde{x}) \| \leq \mu^2 \|x - \tilde{x}\|, \qquad \forall x, \tilde{x} \in \mathbb{R}^n. \tag{2.15}$$

Proof. Setting $y = \mathrm{prox}^D_f(x)$ and $\tilde{y} = \mathrm{prox}^D_f(\tilde{x})$, the following relations are obtained by applying (2.14) to y and $\tilde{y}$, respectively:

$$\begin{aligned} f(z) &\geq f(y) + (z - y)^T D(x - y) \quad \forall \ z \in \mathbb{R}^n \\ f(\tilde{z}) &\geq f(\tilde{y}) + (\tilde{z} - \tilde{y})^T D(\tilde{z} - \tilde{y}) \quad \forall \ \tilde{z} \in \mathbb{R}^n. \end{aligned}$$

Choosing $z = \tilde{y}$, $\tilde{z} = y$ and combining the two inequalities yields

$$\left(x - \operatorname{prox}_f^D(x) - \tilde{x} + \operatorname{prox}_f^D(\tilde{x})\right)^T D\left(\operatorname{prox}_f^D(x) - \operatorname{prox}_f^D(\tilde{x})\right) \geq 0,$$

or equivalently

$$\left\|\operatorname{prox}_f^D(x) - \operatorname{prox}_f^D(\tilde{x})\right\|_D^2 \leq (x - \tilde{x})^T D\left(\operatorname{prox}_f^D(x) - \operatorname{prox}_f^D(\tilde{x})\right).$$

Since $D \in \mathcal{M}_\mu$ and by using the Cauchy-Schwarz inequality, we obtain

$$\left\|\operatorname{prox}_f^D(x) - \operatorname{prox}_f^D(\tilde{x})\right\|^2 \leq \mu^2 \left\|\operatorname{prox}_f^D(x) - \operatorname{prox}_f^D(\tilde{x})\right\| \|x - \tilde{x}\|$$

and thus the thesis holds. □

Proposition 2.13. *Suppose that $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is given by a separable sum of convex functions, i.e.*

$$f(x) = \sum_{i=1}^{r} f_i(x_i),$$

where $f_i : \mathbb{R}^{n_i} \to \bar{\mathbb{R}}$ is proper, convex and lsc for $i = 1, \dots, r$ and $\sum_{i=1}^r n_i = n$. Then

$$\operatorname{prox}_f(x) = \prod_{i=1}^{n} \operatorname{prox}_{f_i}(x_i) = \left(\operatorname{prox}_{f_1}(x_1), \dots, \operatorname{prox}_{f_r}(x_r)\right), \quad \forall\, x \in \mathbb{R}^n.$$

Proof. It follows from (2.14) combined with item (vii) of Proposition 2.7. □

Example 2.10 (ℓ_1–norm)**.** Consider $f(x) = \lambda\|x\|_1$ with $\lambda \in \mathbb{R}_{>0}$, where $\|x\|_1 = \sum_{i=1}^n |x_i|$ is the ℓ_1–norm. Since f is a separable function in $x = (x_1, \dots, x_n)$, the proximal operator of f can be computed component-wise:

$$(\operatorname{prox}_f(x))_i = \operatorname{prox}_{\lambda|\cdot|}(x_i), \quad i = 1, \dots, n.$$

From the equivalence (2.14) we have

$$\begin{aligned} y_i = \operatorname{prox}_{\lambda|\cdot|}(x_i) &\iff x_i - y_i \in \partial(\lambda|\cdot|)(y_i) \\ &\iff y_i = x_i - w_i, \quad w_i \in \partial(\lambda|\cdot|)(y_i) \end{aligned}$$

and by computing the subdifferential $\partial(\lambda|\cdot|)$ we obtain

$$\begin{aligned} (\operatorname{prox}_f(x))_i &= \begin{cases} x_i - \lambda, & \text{if } x_i > \lambda \\ 0, & \text{if } x_i \in [-\lambda, \lambda] \\ x_i + \lambda, & \text{if } x_i < -\lambda \end{cases} \\ &= \operatorname{sign}(x_i)\max\{|x_i| - \lambda, 0\}, \quad i = 1, \dots, n. \end{aligned}$$

This is the so-called *soft-thresholding* (or *shrinkage*) operator.

Proposition 2.14 (Moreau decomposition)**.** *Given a proper, convex, lsc function $f : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$, its conjugate $f^* : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$, $\alpha \in \mathbb{R}_{>0}$ and D a symmetric positive definite matrix, the following identity holds:*

$$\operatorname{prox}_{\alpha f}^{D}(x) + \alpha D^{-1} \operatorname{prox}_{\alpha^{-1} f^*}^{D^{-1}}(\alpha^{-1} D x) = x, \ \forall x \in \mathbb{R}^n.$$

Proof. The Moreau decomposition follows from the properties characterizing the subdifferential and the conjugate of a function. Indeed, given $x \in \mathbb{R}^n$, let $y = \operatorname{prox}_{\alpha f}^{D}(x)$. In the light of equation (2.14) and Proposition 2.4, we obtain

$$\begin{aligned} y = \operatorname{prox}_{\alpha f}^{D}(x) &\iff \alpha^{-1} D(x-y) \in \partial f(y) \\ &\iff y \in \partial f^*(\alpha^{-1} D(x-y)). \end{aligned}$$

By setting $w = \alpha^{-1} D(x-y)$, the last differential inclusion becomes

$$x - \alpha D^{-1} w \in \partial f^*(w)$$

or, equivalently

$$D^{-1}\left(\alpha^{-1} D x - w\right) \in \partial(\alpha^{-1} f^*)(w).$$

Applying again equation (2.14) yields

$$x - y = \alpha D^{-1} \operatorname{prox}_{\alpha^{-1} f^*}^{D^{-1}}(\alpha^{-1} D x)$$

and this concludes the proof. □

Example 2.11 (ℓ_2-norm)**.** Let $f(x) = \lambda\|x\|$ with $\lambda \in \mathbb{R}_{>0}$. By the Moreau decomposition we have

$$\operatorname{prox}_f(x) = x - \operatorname{prox}_{f^*}(x), \quad \forall\, x \in \mathbb{R}^n.$$

From Example 2.2, it is known that $f^* = \iota_{B(0,\lambda)}$. Thus $\operatorname{prox}_{f^*}(x) = P_{B(0,\lambda)}(x) = \lambda x/\|x\|$ and in conclusion

$$\operatorname{prox}_f(x) = \begin{cases} \left(1 - \frac{\lambda}{\|x\|}\right) x, & \text{if } \|x\| > \lambda \\ 0, & \text{if } \|x\| \leq \lambda. \end{cases}$$

Note that, when $n = 1$, the above formula reduces to the scalar soft-thresholding operation seen in Example 2.10.

Example 2.12 (Composite functions)**.** Let $f(x) = g(Ax)$, where $A \in \mathbb{R}^{m \times n}$ is a *semi-orthogonal matrix*, i.e.

$$A^T A = \nu I_n, \quad \nu > 0$$

and $g : \mathbb{R}^m \to \bar{\mathbb{R}}$ is a proper, convex, lsc function. A simple application of equation (2.14) shows that

$$\operatorname{prox}_f^D(x) = \nu^{-1} A^T \operatorname{prox}_{\nu g}^D(Ax).$$

Hence, in this special case, when prox_g^D has a simple closed-form expression, so does prox_f^D. As an example, any function $f(x) = \|Ax\|$ with A semi-orthogonal has an explicit formula for its proximal operator. However, it is important to note that, for a general matrix A, there is no explicit expression of prox_f^D in terms of prox_g^D and A.

2.2 Proximal–gradient methods

The structure of the objective function in (2.1) can be successfully exploited by the class of *proximal–gradient* or *forward–backward* (FB) algorithms [14, 15, 51, 48], whose general iteration is given by

$$x^{(k+1)} = x^{(k)} + \lambda_k \left(\mathrm{prox}_{\alpha_k f_1}(x^{(k)} - \alpha_k \nabla f_0(x^{(k)})) - x^{(k)} \right), \quad k = 0, 1, 2, \dots \tag{2.16}$$

where $\alpha_k \in \mathbb{R}_{>0}$ is a scalar steplength parameter and $\lambda_k \in \mathbb{R}_{\geq 0}$ is the so-called relaxation (or line–search) parameter. At each iteration, the FB method alternates a *forward* gradient step on the differentiable part f_0, followed by a *backward* proximal step on the convex term f_1. Special instances of (2.16) are the following:

- the proximal point algorithm [129] for minimizing a nondifferentiable function f_1, when $f_0 \equiv 0$ and $\lambda_k \equiv 1$:
$$x^{(k+1)} = \mathrm{prox}_{\alpha_k f_1}(x^{(k)});$$
- the steepest descent method, when $f_1 \equiv 0$ and $\lambda_k \equiv 1$;
- the gradient projection method (1.49), when $f_1 = \iota_\Omega$ (see Example 2.9).

Let us remark that any proximal–gradient method involves, at each iteration (2.16), the solution of the convex subproblem related to the evaluation of the proximal operator at the gradient point. Therefore, what happens with the FB method is that the original problem (2.1) is replaced by a sequence of convex subproblems, whose solution needs be known in closed-form or, at least, within a certain accuracy, in order for the method to be effective. In the subsequent discussion and related convergence results, the proximal operator will be always assumed to be known in its exact form.

Before starting with our overview of proximal–gradient methods, we state a fundamental key property for the subsequent convergence analysis and that will be assumed hereafter:

Assumption 2.1. $\nabla f_0 : \mathbb{R}^n \to \mathbb{R}^n$ is $L-$Lipschitz continuous with $L \in \mathbb{R}_{>0}$, i.e.

$$\|\nabla f_0(x) - \nabla f_0(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathbb{R}^n. \tag{2.17}$$

As done in the framework of gradient projection methods, we now distinguish between two possible approaches, in which the linesearch is performed *along the arc* and *along the feasible direction*, respectively.

2.2.1 Along the arc approach

The along the arc approach is obtained by setting $\lambda_k \equiv 1$ in (2.16):

$$x^{(k+1)} = \mathrm{prox}_{\alpha_k f_1}\left(x^{(k)} - \alpha_k \nabla f(x^{(k)})\right). \tag{2.18}$$

There are two straightforward interpretations of algorithm (2.18) that are now reported:

- **Fixed point algorithm:** from Proposition 2.11, we know that a necessary condition for $x^* \in \mathbb{R}^n$ to be a solution of (2.1) is

$$\begin{aligned} x^* &= \operatorname{prox}_{\alpha f_1}(x^* - \alpha \nabla f_0(x^*)) \\ &= (I_n + \alpha \partial f_1)^{-1}(I_n - \alpha \nabla f_0)(x^*) \quad \text{(Proposition 2.12)}. \end{aligned}$$

  Hence x^* is a stationary point of (2.1) if and only x^* is a fixed point for the forward–backward operator $(I_n + \alpha \partial f_1)^{-1}(I_n - \alpha \nabla f_0)$. Then (2.18) can be seen as the sequence generated by the fixed point algorithm applied to $(I_n + \alpha \partial f_1)^{-1}(I_n - \alpha \nabla f_0)$.

- **Quadratic approximation:** the FB iteration (2.18) can also be interpreted as the minimization of a reasonable local approximation of the objective function. Indeed, some algebra shows that

$$\begin{aligned} x^{(k+1)} &= \operatorname{prox}_{\alpha_k f_1}\left(x^{(k)} - \alpha_k \nabla f_0(x^{(k)})\right) \\ &= \operatorname*{argmin}_{x \in \mathbb{R}^n} \frac{1}{2\alpha_k}\|x - (x^{(k)} - \alpha_k \nabla f_0(x^{(k)}))\|^2 + f_1(x) \\ &= \operatorname*{argmin}_{x \in \mathbb{R}^n} \underbrace{f_0(x^{(k)}) + \nabla f_0(x^{(k)})^T(x - x^{(k)}) + \frac{1}{2\alpha_k}\|x - x^{(k)}\|^2}_{:=q_{\alpha_k}(x)} + f_1(x) \qquad (2.19) \\ &= \operatorname*{argmin}_{x \in \mathbb{R}^n} h_{\alpha_k}(x). \qquad (2.20) \end{aligned}$$

  Thus, at each iteration, we see that the function f_0 is being replaced by the local quadratic approximation q_{α_k}, i.e. the linearized part of f_0 regularized by a quadratic proximal term, which measures the local error in the approximation.

Two important istances of the along the arc approach are illustrated by Beck and Teboulle in [14, 15]. As explained by the authors in [15, Section 1.4.2-1.4.3], when the objective function f is convex, the convergence analysis of the along the arc scheme (2.18) is strictly related to the fundamental key property stated below:

$$f(x^{(k+1)}) \leq h_{\alpha_k}(x^{(k+1)}), \quad \forall\, k \in \mathbb{N}. \qquad (2.21)$$

In other words, the steplength α_k must be chosen in such a way that the local approximation h_{α_k} majorizes the approximated function f at the proximal point $x^{(k+1)}$. A simple way to have that is relating the steplength α_k to the Lipschitz constant L of ∇f_0, as suggested by the following Lemma.

Lemma 2.5 (Descent lemma)**.** *Let $f_0 : \mathbb{R}^n \longrightarrow \mathbb{R}$ be a continuously differentiable function satisfying Assumption 2.1. Then*

$$f_0(y) \leq f_0(x) + \nabla f_0(x)^T(y - x) + \frac{L}{2}\|x - y\|^2, \quad \forall\, x, y \in \mathbb{R}^n.$$

Proof. Let $h : \mathbb{R} \to \mathbb{R}$ be such that $h(t) = f_0\big(x + t(y-x)\big)$, for all $t \in \mathbb{R}$. The chain rule yields $\dfrac{dh(t)}{dt} = \nabla f_0\big(x + t(y-x)\big)^T (y-x)$. Moreover, we have

$$\begin{aligned}
f_0(y) - f_0(x) &= h(1) - h(0) = \int_0^1 \frac{dh(t)}{dt}\, dt = \int_0^1 (y-x)^T \nabla f_0\big(x + t(y-x)\big)\, dt \\
&\leq \int_0^1 (y-x)^T \nabla f_0(x)\, dt + \left| \int_0^1 (y-x)^T \big(\nabla f_0\big(x + t(y-x)\big) - \nabla f_0(x)\big)\, dt \right| \\
&\leq \int_0^1 (y-x)^T \nabla f_0(x)\, dt + \int_0^1 \|x-y\| \cdot \|\nabla f_0\big(x + t(y-x)\big) - \nabla f_0(x)\|\, dt \\
&\leq (y-x)^T \nabla f_0(x) + \|x-y\| \int_0^1 Lt\|x-y\|\, dt \\
&= (y-x)^T \nabla f_0(x) + \frac{L}{2}\|x-y\|^2.
\end{aligned}$$

□

A direct consequence of Lemma 2.5 is that condition (2.21) is automatically guaranteed whenever $\alpha_k \in (0, 1/L]$. Then one could decide upon one of the following two strategies:

- if the Lipschitz constant L is known, then

$$\alpha_k = \frac{1}{L}, \quad \forall\, k \in \mathbb{N}. \tag{2.22}$$

  The corresponding method is reported in Algorithm 4 and is sometimes referred to as the *Iterative Soft Thresholding Algorithm (ISTA)*, where the name is borrowed from a special instance of Algorithm 4, which is recovered when $f_1 = \lambda\|\cdot\|_1$ and the proximal operator consequently reduces to the soft-thresholding operator [41, 58].

- if the Lipschitz constant L is not known or cannot be easily computed, such a difficulty may be overcome by performing a linesearch ensuring condition (2.21). In particular, once fixed the values $L_0 \in \mathbb{R}_{>0}$, $\eta > 1$, the parameter α_k is selected as:

$$\alpha_k = \frac{1}{L_k}, \tag{2.23}$$

  where $L_k = \eta^{i_k} L_{k-1}$ and i_k is the smallest nonnegative integer such that

$$f_0(x^{(k+1)}) \leq f_0(x^{(k)}) + (x^{(k+1)} - x^{(k)})^T \nabla f_0(x^{(k)}) + \frac{L_k}{2}\|x^{(k+1)} - x^{(k)}\|^2, \tag{2.24}$$

  where $x^{(k+1)}$ is computed by means of (2.18) combined with (2.23). It should be noted that the above linesearch is well-defined since, thanks to Lemma 2.5, condition (2.24) is always satisfied for $L_k \geq L$. The resulting method, denominated *ISTA with backtracking*, is resumed in Algorithm 5.

For the sake of simplicity, the following notation is used to indicate the proximal operator in Algorithm 4 and 5:

$$p_L(x) = \mathrm{prox}_{\frac{1}{L}f_1}\left(x - \frac{1}{L}\nabla f_0(x)\right).$$

Algorithm 4 ISTA with constant steplengths

```
Choose the starting point $x^{(0)} \in \mathrm{dom}(f_1)$ and let $L \in \mathbb{R}_{>0}$ be the Lipschitz constant of $\nabla f_0$.
FOR $k = 0, 1, 2, \ldots$
        $x^{(k+1)} = p_L(x^{(k)})$.
END
```

Algorithm 5 ISTA with backtracking

```
Choose the starting point $x^{(0)} \in \mathrm{dom}(f_1)$ and let $L_{-1} \in \mathbb{R}_{>0}$, $\eta > 1$.
FOR $k = 0, 1, 2, \ldots$
    STEP 1. Compute the smallest nonnegative integer $i_k$ such that $L_k = \eta^{i_k} L_{k-1}$ satisfies
        $f_0(p_{L_k}(x^{(k)})) \leq f_0(x^{(k)}) + (p_{L_k}(x^{(k)}) - x^{(k)})^T \nabla f_0(x^{(k)}) + \frac{L_k}{2}\|p_{L_k}(x^{(k)}) - x^{(k)}\|^2$.
    STEP 2. Compute $x^{(k+1)} = p_{L_k}(x^{(k)})$.
END
```

Remark 2.8. The sequence of function values $\{f(x^{(k)})\}_{k\in\mathbb{N}}$ produced both by ISTA and ISTA with backtracking is nonincreasing. In fact, by choosing L_k with the backtracking rule (2.24) or $L_k \equiv L$, we have:

$$f(x^{(k+1)}) \leq h_{1/L_k}(x^{(k+1)}) \leq h_{1/L_k}(x^{(k)}) = f(x^{(k)})$$

where the first inequality follows from STEP 1 of Algorithm 5 and the second one is a consequence of the definition of proximal point (see the quadratic approximation interpretation).

Remark 2.9. Since (2.24) holds for $L_k \geq L$, then for the ISTA with backtracking it holds that $L_k \leq \eta L$ for every $k \geq 1$, so that overall

$$\beta L \leq L_k \leq \gamma L,$$

where $\beta = \gamma = 1$ for the constant steplength setting and $\beta = \dfrac{L_{-1}}{L}, \gamma = \eta$ for the backtracking case.

Remark 2.10. The linesearch procedure (2.24) avoids the difficulty of knowing the Lipschitz constant only partially. Indeed, even if not explicitly, the parameter L depends on the value of the initial guess L_{-1} by relation $L_k = (\prod_{j=1}^{k} \eta^{i_j})L_{-1}$. Therefore, a wrong choice of the initial guess L_{-1} might negatively affect the convergence rate of the whole algorithm. For instance, if L_{-1} is fixed very far from the unknown Lipschitz value, this could lead to either a very small steplength (if L_{-1} is too large) or a great number of successive linesearch reductions (if it is too small).
More in general, a major critical issue of any backtracking procedure applied to the scheme (2.18) is that a new evaluation of the proximal operator is required at any iteration of the backtracking loop. For that reason, the along the arc approach becomes computationally too expensive if the proximal point cannot be computed in a reasonable time.

The following Theorem states, under the assumption that f_0 is convex, the convergence of the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ generated by the two ISTA methods to a solution of problem (2.1), and a sublinear rate of convergence for their function values.

Theorem 2.2. *Let $f : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ be as in problem (2.1), where f_0 is convex, continuously differentiable and satisfies Assumption 2.1, and f_1 is proper, convex and lower semicontinuous. Suppose that (2.1) admits at least one solution. Let $\{x^{(k)}\}_{k\in\mathbb{N}}$ be the sequence generated by Algorithm 4 or 5. Then*

(i) the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ converges to a solution of problem (2.1).

(ii) For every $k \geq 1$:

$$f(x^{(k)}) - f(x^*) \leq \frac{\gamma L\|x^{(0)} - x^*\|^2}{2k}$$

for any optimal solution x^*.

Proof. See [15, Theorem 1.1-1.2]. □

We now report a convergence result for ISTA when f is nonconvex, which is of course weaker than the one presented in Theorem 2.2 for the convex case.

Theorem 2.3. *Let $f : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ be as in problem (2.1), where f_0 is continuously differentiable and satisfies Assumption 2.1, and f_1 is proper, convex and lower semicontinuous. Let $\{x^{(k)}\}_{k\in\mathbb{N}}$ be the sequence generated by Algorithm 4 or 5. Then*

$$\lim_{k\to+\infty} \|x^{(k)} - x^{(k+1)}\| = 0.$$

Proof. See [15, Theorem 1.3]. □

Remark 2.11. If we observe that $\|x^{(k)} - x^{(k+1)}\| = \|x^{(k)} - \mathrm{prox}_{1/L_k}(x^{(k)} - 1/L_k\nabla f_0(x^{(k)}))\|$ and that a point x^* is stationary for f if and only $x^* = \mathrm{prox}_{1/L}(x^* - 1/L\nabla f_0(x^*))$ for any fixed $L \in \mathbb{R}_{>0}$ (Proposition 2.11), then $\|x^{(k)} - x^{(k+1)}\|$ can be considered as a measure of the proximity of the sequence to a stationary point which, by Theorem 2.3, is converging to zero.

2.2.2 Along the feasible direction approach

We now explore the case in which a relaxation parameter λ_k is introduced in the scheme (2.16). In this case, the steplength α_k is usually chosen either by an adaptive selection rule or a prefixed formula, while the parameter λ_k is determined via a backtracking procedure of some sort.

The seminal work by Combettes [51] suggested a scheme in which the steplengths are variable but strictly depending on the value of the Lipschitz constant in accordance with the following condition:

$$0 < \inf_{k\in\mathbb{N}} \alpha_k \leq \sup_{k\in\mathbb{N}} \alpha_k < \frac{2}{L}, \tag{2.25}$$

whereas the relaxation parameter is bounded above by 1 and bounded away from zero

$$0 < \inf_{k\in\mathbb{N}} \lambda_k \leq \sup_{k\in\mathbb{N}} \lambda_k \leq 1. \tag{2.26}$$

A special instance of this scheme has been proposed by Combettes and Pesquet in [48] and is reported in Algorithm 6.

Algorithm 6 Forward-backward method with relaxation parameters and variable steplengths

```
Choose the starting point $x^{(0)} \in \mathrm{dom}(f_1)$, let $L \in \mathbb{R}_{>0}$ be the Lipschitz constant of $\nabla f_0$ and fix $\epsilon \in (0, \min\{1, 1/L\})$.
FOR $k = 0, 1, 2, \ldots$
    STEP 1. Choose $\alpha_k \in [\epsilon, \frac{2}{L} - \epsilon]$.
    STEP 2. Compute $y^{(k)} = \mathrm{prox}_{\alpha_k f_1}\left(x^{(k)} - \alpha_k \nabla f_0(x^{(k)})\right)$.
    STEP 3. Choose $\lambda_k \in [\epsilon, 1]$.
    STEP 4. Compute $x^{(k+1)} = x^{(k)} + \lambda_k(y^{(k)} - x^{(k)})$.
END
```

Convergence may be proved in the convex case, as stated by the following result.

Theorem 2.4. *[51, Theorem 3.4] Suppose that f_0 in problem* (2.1) *is convex, continuously differentiable and satisfies Assumption 2.1, and f_1 is proper, convex and lower semicontinuous. Every sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ generated by Algorithm 6 or, more generally, by any method of type* (2.16) *satisfying conditions* (2.25)-(2.26)*, converges to a solution of problem* (2.1)*.*

Algorithm 6 features variable steplengths, but its relaxation parameters $\{\lambda_k\}_{k\in\mathbb{N}}$ are not allowed to exceed 1. The variant proposed in [12] and resumed in Algorithm 7 allows for larger relaxation parameters, at the price of keeping fixed the steplength parameter.

Theorem 2.5. *[12] Suppose that f_0 in problem* (2.1) *is convex, continuously differentiable and satisfies Assumption 2.1, and f_1 is proper, convex and lower semicontinuous. Every sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ generated by Algorithm 7 converges to a solution of problem* (2.1)*.*

Algorithm 7 Forward-backward method with relaxation parameters and constant steplengths

```
Choose the starting point $x^{(0)} \in \mathrm{dom}(f_1)$, let $L \in \mathbb{R}_{>0}$ be the Lipschitz constant of $\nabla f_0$ and
fix $\epsilon \in (0, 3/4)$.
FOR $k = 0, 1, 2, \dots$
    STEP 1. Compute $y^{(k)} = \mathrm{prox}_{\frac{1}{L}f_1}\left(x^{(k)} - \frac{1}{L}\nabla f_0(x^{(k)})\right)$.
    STEP 2. Choose $\lambda_k \in [\epsilon, \frac{3}{2} - \epsilon]$.
    STEP 3. Compute $x^{(k+1)} = x^{(k)} + \lambda_k(y^{(k)} - x^{(k)})$.
END
```

Remark 2.12. The applicability of Algorithm 6 and 7 is limited to the cases in which the Lipschitz constant is explicitly computable. However, there is a number of problems, arising from signal and image processing, in which the knowledge of the Lipschitz constant is out of reach. For instance, the Kullback-Leibler divergence with positive background (see Chapter 5), which arises in the context of image denoising with data corrupted by Poisson noise, has a Lipschitz continuous gradient, but only an above estimation of the Lipschitz parameter is available [82]. Furthermore, to the best of my knowledge, no practical selection rule or line–search to determine the relaxation parameter according to STEP 3 of Algorithm 6 or STEP 2 of Algorithm 7 has been proposed in the literature yet.

2.2.3 Acceleration strategies

Though appealing for their simplicity, proximal–gradient methods often exhibit a slow speed of convergence. This is a common problem shared by all first order methods, both in the differentiable and nondifferentiable setting. In the literature, two significant strategies have been devised to accelerate forward–backward schemes: adding an extrapolation step and adopting a variable metric in the computation of the proximal operator.

Inertial/Extrapolation techniques

Extrapolation in gradient methods was first introduced by Polyak in [106], where he studied the well-known *Heavy-Ball method* for minimizing strongly convex functions with Lipschitz continuous gradient:

$$x^{(k+1)} = x^{(k)} - \alpha\nabla f(x^{(k)}) + \beta(x^{(k)} - x^{(k-1)})$$

with $\alpha \in \mathbb{R}_{>0}$, $\beta \in [0, 1)$. This iterative scheme can be seen as an explicit finite differences discretization of the so-called *Heavy-ball with friction* dynamical system

$$\ddot{x}(t) + \gamma\dot{x}(t) + \nabla f(x(t)) = 0$$

which arises when Newton's law is applied to a point subject to a constant friction $\gamma \in \mathbb{R}_{>0}$ and a gravity potential f. The term $\beta(x^{(k)} - x^{(k-1)})$ is usually referred to as the *inertial force* or *extrapolation step*, and introduces information about the two previous iterates. Note that setting $\beta = 0$ returns the usual gradient method. The surprising fact about the Heavy-Ball method is that, with a negligible additional overhead due to the extrapolation step, it provides an optimal $\mathcal{O}(1/k^2)$ rate for strongly convex functions [106].

The extrapolation idea can be easily transposed in the context of forward–backward algorithms. Ochs et al [105] recently proposed a generalization of the Heavy-Ball method to problem (2.1), denominated *inertial Proximal algorithm for Nonconvex Optimization (iPiano)*, of the following type

$$x^{(k+1)} = \mathrm{prox}_{\alpha_k f_1}\left(x^{(k)} - \alpha_k \nabla f_0(x^{(k)}) + \beta_k(x^{(k)} - x^{(k-1)})\right) \tag{2.27}$$

where the variable parameters α_k and β_k must be appropriately chosen in order to make the algorithm convergent. To have an idea of what could be an appropriate choice for the iPiano parameters, we report, in Algorithm 8, one of the several versions of iPiano delineated by the authors, which keeps fixed β while determining α_k by means of a backtracking procedure.

Algorithm 8 Nonmonotone iPiano

```
Choose $x^{(0)} \in \mathrm{dom}(f_1)$, $L_{-1} \in \mathbb{R}_{>0}$, $\eta \geq 1$, $\beta \in [0,1)$ and set $x^{(-1)} = x^{(0)}$.
FOR $k = 0, 1, 2, \ldots$

  STEP 1. Set $L_k = L_{k-1}$.

  STEP 2. Backtracking loop:
            Choose $\alpha_k < 2(1-\beta)/L_k$ and compute

              $x^{(k,L_k)} = \mathrm{prox}_{\alpha_k f_1}\left(x^{(k)} - \alpha_k \nabla f_0(x^{(k)}) + \beta(x^{(k)} - x^{(k-1)})\right)$.

            IF $f_0(x^{(k,L_k)}) \leq f_0(x^{(k)}) + (x^{(k,L_k)} - x^{(k)})^T \nabla f_0(x^{(k)}) + \frac{L_k}{2}\|x^{(k,L_k)} - x^{(k)}\|^2$ THEN
                go to STEP 3
            ELSE
                set $L_k = \eta L_k$ and go to STEP 2.
            ENDIF

  STEP 3. Compute $x^{(k+1)} = x^{(k,L_k)}$.

END
```

The convergence of iPiano can be proved in the nonconvex case, under the assumption that the function f satisfies the so-called *Kurdyka–Lojasiewicz inequality* (see Section 2.3 for an overview of this property).

Algorithm 9 FISTA with backtracking

```
Choose $x^{(0)} \in \mathrm{dom}(f_1)$, $L_{-1} \in \mathbb{R}_{>0}$, $\eta > 1$, $a > 2$. Set $y^{(0)} = x^{(0)}$, $t_0 = 1$.
FOR $k = 0, 1, 2, \ldots$
    STEP 1. Compute the smallest nonnegative integer $i_k$ such that $L_k = \eta^{i_k} L_{k-1}$ satisfies
        $$f_0(p_{L_k}(y^{(k)})) \leq f_0(y^{(k)}) + (p_{L_k}(y^{(k)}) - y^{(k)})^T \nabla f_0(y^{(k)}) + \frac{L_k}{2}\|p_{L_k}(y^{(k)}) - y^{(k)}\|^2.$$
    STEP 2. Compute $x^{(k+1)} = p_{L_k}(y^{(k)})$.
    STEP 3. Compute $t_{k+1} = \frac{k+a}{a}$.
    STEP 4. Compute $y^{(k+1)} = x^{(k)} + \left(\frac{t_k - 1}{t_{k+1}}\right)(x^{(k)} - x^{(k-1)})$.
END
```

An ingenious variant of the Heavy-Ball method, that was initially treated by Nesterov in [102] for gradient methods and subsequently extended to proximal–gradient methods, is the following

$$\begin{aligned} y^{(k)} &= x^{(k)} + \beta_k(x^{(k)} - x^{(k-1)}) \\ x^{(k+1)} &= x^{(k)} - \alpha \nabla f(y^{(k)}). \end{aligned}$$

Here we highlight two main changes with respect to the Heavy-Ball method: the extrapolation factor β_k is variable and computed according to a prefixed formula and, at each iteration, the gradient is evaluated at the extrapolated point $y^{(k)}$ instead of $x^{(k)}$. The resulting algorithm is still optimal, showing an $\mathcal{O}(1/k^2)$ complexity result.

It is then natural to extend the aforementioned extrapolated scheme to proximal–gradient methods:

$$\begin{aligned} y^{(k)} &= x^{(k)} + \beta_k(x^{(k)} - x^{(k-1)}) \\ x^{(k+1)} &= \mathrm{prox}_{\alpha_k f_1}(x^{(k)} - \alpha \nabla f_0(y^{(k)})). \end{aligned}$$

The combination of Nesterov acceleration technique with the proximal–gradient method ISTA led to the rise of the popular *Fast Iterative Soft Thresholding Algorithm (FISTA)* [14, 42] for solving problem (2.1) (see Algorithm 9). In FISTA, the parameter β_k is chosen as

$$\beta_k = \frac{t_k - 1}{t_{k+1}}$$

where $t_k \geq 1$, for all $k \in \mathbb{N}$. The original choice of Beck and Teboulle in [14], namely $t_{k+1} = \left(1 + \sqrt{1 + 4t_k^2}\right)/2$, ensures an $\mathcal{O}(1/k^2)$ convergence rate for FISTA, which improves the result contained in Theorem 2.2 for ISTA. However, such a choice for t_k does not guarantee the convergence of the iterates $\{x^{(k)}\}_{k\in\mathbb{N}}$. For this reason, in Algorithm 9 the parameter t_k is computed following the rule suggested by Chambolle and Dossal in [42], which allows to prove the weak convergence of the algorithm in general Hilbert spaces (which in $\mathbb{R}^n$ is equivalent to the strong convergence of the sequence) while at the same time preserving the $\mathcal{O}(1/k^2)$ complexity result. The result is reported below in the finite dimensional case.

Theorem 2.6. *Let $f : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ be as in problem* (2.1)*, where f_0 is convex, continuously differentiable and satisfies Assumption 2.1, and f_1 is proper, convex and lower semicontinuous. Suppose that* (2.1) *admits at least one solution. Let $\{x^{(k)}\}_{k\in\mathbb{N}}$ be the sequence generated by Algorithm 9. Then*

(i) the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ converges to an optimal solution of problem (2.1)*.*

(ii) For every $k \geq 1$:

$$f(x^{(k)}) - f(x^*) \leq \frac{2\gamma L\|x^{(0)} - x^*\|^2}{(k+1)^2}$$

for any optimal solution x^*.

Proof. See [42, Theorem 3]. □

Variable metric techniques

If a variable metric in the computation of the proximity operator is introduced in (2.16), the general iteration of the FB scheme becomes

$$x^{(k+1)} = x^{(k)} + \lambda_k \left(\mathrm{prox}^{D_k}_{\alpha_k f_1}(x^{(k)} - \alpha_k D_k^{-1}\nabla f_0(x^{(k)})) - x^{(k)}\right), \quad k = 0, 1, 2, \dots \tag{2.28}$$

where D_k is a symmetric positive definite matrix. We call this modified scheme the *Variable Metric Forward Backward* algorithm. The role of the scaling matrix D_k is better appreciated if, as already done in Section 2.2.1 for the case $D_k = I_n$, we interpret the variable metric forward–backward step as the minimization of a local approximation of f at the iterate $x^{(k)}$:

$$\begin{aligned}
y^{(k)} &= \mathrm{prox}^{D_k}_{\alpha_k f_1}\left(x^{(k)} - \alpha_k D_k^{-1}\nabla f_0(x^{(k)})\right) \\
&= \underset{y\in\mathbb{R}^n}{\mathrm{argmin}}\, \frac{1}{2\alpha_k}\left\|y - (x^{(k)} - \alpha_k D_k^{-1}\nabla f_0(x^{(k)}))\right\|^2_{D_k} + f_1(y) \\
&= \underset{y\in\mathbb{R}^n}{\mathrm{argmin}}\, \nabla f_0(x^{(k)})^T(y - x^{(k)}) + \frac{1}{2\alpha_k}\|y - x^{(k)}\|^2_{D_k} + \frac{\alpha_k}{2}\|\nabla f_0(x^{(k)})\|^2_{D_k^{-1}} + f_1(y) \\
&= \underset{x\in\mathbb{R}^n}{\mathrm{argmin}}\, \underbrace{f_0(x^{(k)}) + \nabla f_0(x^{(k)})^T(y - x^{(k)}) + \frac{1}{2\alpha_k}\|y - x^{(k)}\|^2_{D_k}}_{:=q(x,x^{(k)})} + f_1(y) \\
&= \underset{x\in\mathbb{R}^n}{\mathrm{argmin}}\, h^{(k)}(x, x^{(k)}).
\end{aligned}$$

Therefore, in order for VMFB to be an effective tool, the matrix D_k has to be chosen in such a way that the quadratic model $q(x, x^{(k)})$ represents a better approximation than the one defined by the FB method in (2.19) without variable metric. For instance, this is the case when f_0 is twice continuously differentiable and D_k approximates the Hessian matrix $\nabla^2 f_0(x^{(k)})$, so that the quadratic term $q(x, x^{(k)})$ is close to the second order Taylor expansion of the function f_0 at point $x^{(k)}$. Then the issue of devising practical techniques to compute the matrix D_k arises. In this regard, a couple of choices proposed in the literature for the matrix D_k are now reported:

- **Convergent variable metric:** in [50], weak convergence of the sequence generated by the VMFB method in a general Hilbert space, under the hypothesis that the differentiable term f_0 is convex, is provided when the sequence $\{D_k\}_{k\in\mathbb{N}}$ satisfies the following two conditions

$$\begin{aligned} &D_k \in \mathcal{M}_\mu, \\ &(1+\xi_k)D_{k+1} \succcurlyeq D_k, \quad \xi_k \in \mathbb{R}_{\geq 0}, \ \sum_{i=1}^{\infty} \xi_k < +\infty \end{aligned} \tag{2.29}$$

  where " $\succcurlyeq$" denotes the Loewner partial ordering on the set of all symmetric matrices, i.e.

$$A \succcurlyeq B \iff x^T A x \geq x^T B x, \quad \forall\ x \in \mathbb{R}^n.$$

  According to Lemma 2.3 in [49], condition (2.29) implies that the sequence $\{D_k\}_{k\in\mathbb{N}}$ is converging to a certain symmetric positive definite matrix D, namely

$$D_k x \ \longrightarrow\ Dx, \quad \forall\ x \in \mathbb{R}^n.$$

- **Majorize-Minimize metric:** another possible choice for the matrix D_k is given by the Majorize-Minimize (MM) strategy [47], according to which the matrix D_k is chosen in such a way that the local quadratic model

$$q(x, x^{(k)}) = f_0(x^{(k)}) + \nabla f_0(x^{(k)})^T (x - x^{(k)}) + \frac{1}{2}\|x - x^{(k)}\|^2_{D_k}$$

  is a *majorant function* for f_0 at $x^{(k)}$, i.e.

$$f_0(x) \leq q(x, x^{(k)}), \quad \forall x \in \mathbb{R}^n. \tag{2.30}$$

2.3 Inexact proximal–gradient methods under the Kurdyka–Łojasiewicz property

Recent works [9, 27, 28, 66] have shed light on an interesting analytical property shared by a large variety of functions, namely the so-called *Kurdyka-Łojasiewicz (KL) inequality.* The interest in this property is twofold: on one hand, it is possible to link the convergence of inexact descent methods (thus, in particular, of proximal–gradient algorithms) to this property, provided that the aforementioned methods comply with some specific conditions [9, 66]; on the other hand, many problems frequently addressed in signal and image processing involve functions satisfying the KL inequality [8, 27], which highlights the general validity of the convergence result under this property.

2.3.1 Kurdyka–Łojasiewicz (KL) functions

The following definition of KL property is the one employed also in [8, 9, 105], but we remark that other versions of this property have been considered in the literature [7, 29, 47].

Definition 2.13. *Let $f : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ be a proper, lower semicontinuous function. The function f is said to have the* Kurdyka–Łojasiewicz (KL) property *at $\overline{z} \in \mathrm{dom}(\partial f)$ if there exist $\upsilon \in (0, +\infty]$, a neighborhood U of $\overline{z}$ and a continuous concave function $\phi : [0, \upsilon) \longrightarrow [0, +\infty)$ such that:*

- $\phi(0) = 0$*;*
- *ϕ is C^1 on $(0, \upsilon)$;*
- *$\phi'(s) > 0$ for all $s \in (0, \upsilon)$;*
- *the KL inequality*
$$\phi'(f(z) - f(\overline{z}))\mathrm{dist}(0, \partial f(z)) \geq 1$$
holds for all $z \in U \cap [f(\overline{z}) < f < f(\overline{z}) + \upsilon]$.

If f satisfies the KL property at each point of $\mathrm{dom}(\partial f)$*, then f is called a* Kurdyka–Łojasiewicz (KL) function*.*

Remark 2.13. The KL property holds for any non stationary point $\overline{z}$, i.e. such that $0 \notin \partial f(\overline{z})$. Indeed, as explained in [8], in this case there exist $\epsilon, \eta \in \mathbb{R}_{>0}$ and c such that

$$\mathrm{dist}(0, \nabla f(z)) \geq c > 0$$

for all $z \in B(\overline{z}, \epsilon) \cap [-\eta < f < \eta]$, that is, f has the Kurdyka-Łojasiewicz property at $\overline{z}$ with $\phi(t) = c^{-1}t$. Thus, the KL property becomes relevant and non trivial only when it is satisfied at stationary points.

Remark 2.14. When f is differentiable, finite-valued and $f(\overline{z}) = 0$, then the KL property can be rewritten as

$$\|\nabla(\phi \circ f)(z)\| \geq 1$$

for each convenient $z \in \mathbb{R}^n$. In other words, the function f is "sharp" up to a reparametrization of the values f via ϕ. The function ϕ is called *desingularizing*, since it is used to turn a singular region, namely a region in which the gradients are arbitrarily small, into a regular region, i.e. a neighbourhood of $\overline{z}$ where the gradients are bounded away from zero.

Some examples of functions for which the KL property either holds everywhere or fails at certain stationary points are now reported.

Example 2.13 (Real analytic functions)**.** In [93], Łojasiewicz proved that any real analytic function $f : \Omega \to \mathbb{R}$ defined on a nonempty open subset Ω of $\mathbb{R}^n$ satisfies Definition 2.13 at any point $\overline{z} \in \Omega$ with $\phi(t) = cs^{1-\theta}$, where $\theta \in [1/2, 1)$ and $c \in \mathbb{R}_{>0}$. In this case, the KL inequality reduces to

$$\frac{|f(z) - f(\overline{z})|^\theta}{\|\nabla f(z)\|} \leq c$$

on a neighbourhood of $\overline{z}$.

Example 2.14 (C^∞ counterexample)**.** Consider the function $f : \mathbb{R} \to \mathbb{R}$ such that

$$f(x) = \begin{cases} x^2 \sin(\frac{1}{x}), & \text{if } x \neq 0 \\ 0, & \text{if } x = 0. \end{cases}$$

This function is $C^\infty(\mathbb{R})$, however it does not satisfy the KL inequality at the stationary point $x = 0$. Indeed, there exists a sequence $\{x^{(k)}\}_{k\in\mathbb{N}} \subseteq \mathbb{R}$ such that $x^{(k)} \to 0$ and $f'(x^{(k)}) = 0$ for all $k \in \mathbb{N}$, hence there is no ϕ nor U such that the KL inequality holds.

More in general, the KL property fails whenever the considered stationary point is not isolated. See [27, 29] for other examples of the same kind.

Example 2.15 (Locally strongly convex functions)**.** Consider a function $f : \mathbb{R}^n \to \mathbb{R}$ which is strongly convex with modulus μ on K convex subset of $\mathbb{R}^n$, i.e.

$$f(y) \geq f(x) + v^T(y - x) + \frac{\mu}{2}\|y - x\|^2, \quad \forall\, v \in \partial f(x),\ \forall\, x, y \in K.$$

Rearranging the definition, we have

$$\begin{aligned} f(y) - f(x) &\geq v^T(y - x) + \frac{\mu}{2}\|x - y\|^2 \\ &\geq -\frac{1}{\mu}\|v\|^2, \quad \forall\, v \in \partial f(x) \end{aligned}$$

where the last inequality follows by minimizing the middle term over y. Thus

$$\mu(f(x) - f(y)) \leq (\operatorname{dist}(0, \partial f(x)))^2$$

and f satisfies the KL inequality at any point $y \in K$ with $\phi(t) = \frac{2}{\mu}\sqrt{t}$ and $U = K \cap \{x \in \mathbb{R}^n : f(x) \geq f(y)\}$.

Example 2.16 (Convex counterexample)**.** There exists a C^2 convex function $f : \mathbb{R}^2 \to \mathbb{R}$ such that $\min f = 0$ which is not a KL function. More precisely, for each $\upsilon > 0$ and ϕ satisfying the properties in Definition 2.13, it holds

$$\inf\{\|\nabla(\phi \circ f(x))\| :\ x \in [0 < f < \upsilon]\} = 0.$$

Such counterexample can be found in [29]. Note that the considered convex function exhibits a wildly oscillatory collection of level sets that are unlikely to appear in most convex functions.

Let us report a few wide classes of functions that satisfy the KL property in each point of their domain.

Sub-analytic and semi-algebraic functions

Definition 2.14. *(i) A subset $S \subseteq \mathbb{R}^n$ is called* sub-analytic *if each point of S admits a neighborhood V and $m \geq 1$ such that $A \cap V = \{v \in \mathbb{R}^n : \ (x, y) \in B, \ \forall \ y \in \mathbb{R}^m\}$, where B is bounded and*

$$B = \bigcup_{i=1}^{p} \bigcap_{j=1}^{q} \{v \in \mathbb{R}^n : \ g_{ij}(v) = 0, \ h_{ij}(v) < 0\}$$

where the functions g_{ij}, $h_{ij} : \mathbb{R}^n \to \mathbb{R}$ are real analytic for all $1 \leq i \leq p$, $1 \leq j \leq q$.

(ii) A function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is called sub-analytic *if its graph is a sub-analytic subset of $\mathbb{R}^n \times \mathbb{R}$.*

Bolte [27] was able to show that the KL property holds for any sub-analytic continuous function defined on a closed domain, as reported in the following theorem.

Theorem 2.7. *[27, Theorem 3.1] Let $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ be a proper and lower semicontinuous function. If* $\mathrm{dom}(f_1)$ *is closed and f is sub-analytic and continuous on* $\mathrm{dom}(f_1)$*, then it satisfies the KL property at any point of* $\mathrm{dom}(f)$ *with $\phi(t) = ct^{1-\theta}$, where $c \in \mathbb{R}_{>0}$ and $\theta \in [0, 1)$.*

Examples of sub-analytic functions are real analytic functions, for which Theorem 2.7 recovers the same result obtained by Łojasiewicz in [93], and semi-algebraic functions, whose definition is reported below.

Definition 2.15. *(i) A subset $S \subseteq \mathbb{R}^n$ is a* real semi-algebraic set *if*

$$S = \bigcup_{i=1}^{p} \bigcap_{j=1}^{q} \{v \in \mathbb{R}^n : \ g_{ij}(v) = 0, \ h_{ij}(v) < 0\}.$$

where the functions g_{ij}, $h_{ij} : \mathbb{R}^n \to \mathbb{R}$ are real polynomials for all $1 \leq i \leq p$, $1 \leq j \leq q$.

(ii) A function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is called semi-algebraic *if its graph* $\mathrm{graph}(f) = \{(v, t) \in \mathbb{R}^{n+1} : h(v) = t\}$ *is a semi-algebraic subset of* $\mathbb{R}^{n+1}$.

In other words, a function is semi-algebraic whenever its graph is given by finite unions and intersections of polynomial equalities and inequalities.

Example 2.17. The following functions are all semi-algebraic [26]:

- real polynomial functions;

- indicator functions of semi-algebraic sets (such as polyhedral sets);
- finite sums and products or compositions of semi-algebraic functions.

Example 2.18 ($p-$norms)**.** Given $p > 0$, the $p-$norm is defined as

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}, \quad \forall\, x \in \mathbb{R}^n.$$

If p is rational, i.e. $p = \frac{p_1}{p_2}$ where p_1 and p_2 are positive integers, then $\|\cdot\|_p$ is semi-algebraic. Since the sum and composition of semi-algebraic functions is itself semi-algebraic, we just need to prove the statement for the function $g(s) = s^{p_1/p_2}$, for all $s > 0$. Its graph in $\mathbb{R}^2$ can be written as

$$\left\{(s,t) \in \mathbb{R}^2_{>0} :\ t = s^{\frac{p_1}{p_2}}\right\} = \left\{(s,t) \in \mathbb{R}^2 :\ t^{p_2} - s^{p_1} = 0\right\} \cap \mathbb{R}^2_{>0}.$$

This last set is semi-algebraic by Definition 2.15.

On the other hand, $\|\cdot\|_p$ fails to be semi-algebraic whenever p is irrational.

Example 2.19. The last example, combined with the fact that compositions, sums and products of semi-algebraic functions are still semi-algebraic, implies that the following *regularized least squares term*

$$f(x) = \|Ax - b\|^2 + \lambda \sum_{i=1}^{r} \|L_i x\|_p^p + \iota_\Omega(x) \tag{2.31}$$

with $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, $\lambda \in \mathbb{R}_{>0}$, $p \in \mathbb{Q}_{>0}$, $L_i \in \mathbb{R}^{m_i\times n}$ and $\Omega \subseteq \mathbb{R}^n$ a semi-algebraic set, is itself a semi-algebraic function. Hence, thanks to Theorem 2.7, the following optimization problems arising in signal and image processing may be included in the KL framework:

- **Tiknonov regularization:** $\min\limits_{x\in\mathbb{R}^n} \|Ax - b\|^2 + \lambda\|Lx\|^2, \quad$ with $L \in \mathbb{R}^{n\times n}$.
- **ℓ_1 regularization:** $\min\limits_{x\in\mathbb{R}^n} \|Ax - b\|^2 + \lambda\|x\|_1$.
- **Wavelet-based regularization:** $\min\limits_{x\in\mathbb{R}^n} \|Ax-b\|^2+\lambda\|Wx\|_1$, where $W \in \mathbb{R}^{n\times n}$ is a wavelet transform matrix.
- **TV-based regularization:** $\min\limits_{x\in\mathbb{R}^n} \|Ax-b\|^2 + \lambda \sum_{i=1}^n \|\nabla_i x\|$, where $\nabla_i \in \mathbb{R}^{2\times n}$ represents the discrete gradient of the two dimensional object x at pixel i.
- **Constrained regularization:** $\min\limits_{x\in\Omega} \|Ax - b\|^2$, where Ω may be either the nonnegative orthant, a linear equality constraint or the intersection of the two previous constraints.

Sum of real analytic and semi-algebraic functions

According to [26], if a function f is given by $f = f_0 + f_1$ where

- f_0 and f_1 are both sub-analytic,

- f_0 maps bounded sets into bounded sets,

then f is sub-analytic and, by Theorem 2.7, this means that f satisfies the KL property at each point of its domain. For instance, the above conditions are satisfied when f_0 is real analytic and f_1 is semi-algebraic. This latter case is of particular interest, since several nonconvex objective functions arising in image processing are given by the sum of a real analytic function and a semi-algebraic term. The reader may consult Chapter 4 and 5 for several examples of objective functions included in this class.

2.3.2 An abstract convergence result for inexact descent methods

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a proper, lower semicontinuous function that satisfies the Kurdyka-Łojasiewicz property at each point of its domain. In [9], the authors prove an abstract convergence result for any sequence $\{x^{(k)}\}_{k\in\mathbb{N}} \subseteq \mathbb{R}^n$ satisfying the following assumptions:

(C1) (*Sufficient decrease condition*) For some $a \in \mathbb{R}_{>0}$, for all $k \in \mathbb{N}$

$$f(x^{(k+1)}) + a\|x^{(k+1)} - x^{(k)}\|^2 \leq f(x^{(k)});$$

(C2) (*Relative error condition*) For some $b \in \mathbb{R}_{>0}$, for all $k \in \mathbb{N}$, there exists $v^{(k+1)} \in \partial f(x^{(k+1)})$ such that

$$\|v^{(k+1)}\| \leq b\|x^{(k+1)} - x^{(k)}\|;$$

(C3) (*Continuity condition*) There exists a subsequence $\{x^{k_j}\}_{j\in\mathbb{N}}$ and $\tilde{x}$ such that

$$x^{k_j} \to \tilde{x} \text{ and } f(x^{k_j}) \to f(\tilde{x}).$$

Condition (C1) models a sufficient descent property in the function values, while (C2) originates from the fact that most algorithms in optimization, including gradient and proximal–gradient methods, generate an infinite sequence of minimization subproblems which often need to be solved *inexactly*; thus (C2) expresses an inexactness condition for such subproblems. Note also that (C3) is trivial only when f is continuous on its domain.

Theorem 2.8. *[8, Theorem 2.9] Let $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ be a proper, lower semicontinuous function, and consider a sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ satisfying (C1)-(C3). If f has the KL property at the limit point $\tilde{x}$ specified in (C3), then the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ converges to $\tilde{x}$ and $\tilde{x}$ is stationary for f. Furthermore, the sequence has finite length, i.e.*

$$\sum_{k=0}^{\infty} \|x^{(k+1)} - x^{(k)}\| < +\infty.$$

Conditions (C1)-(C3) provide a general scheme for proving the convergence of inexact descent methods in the nonconvex case. In particular, recent works [9, 47] have devised new proximal–gradient algorithms which fit into this framework:

- in [9], under the assumption that the gradient of the differentiable part f_0 is $L-$Lipschitz continuous and without asking for the convexity of f_1, the authors propose an inexact version of the classical forward–backward algorithm, in which both the descent condition and the optimality conditions for the proximal point are relaxed. This inexact version must comply with the following requirements:

$$f_1(x^{(k+1)}) + \nabla f_0(x^{(k)})^T(x^{(k+1)} - x^{(k)}) + \frac{a}{2}\|x^{(k+1)} - x^{(k)}\|^2 \leq f_1(x^{(k)}) \tag{2.32}$$

$$v^{(k+1)} \in \partial f_1(x^{(k+1)}) \tag{2.33}$$

$$\|v^{(k+1)} + \nabla f_0(x^{(k)})\| \leq b\|x^{(k+1)} - x^{(k)}\| \tag{2.34}$$

where $a, b \in \mathbb{R}_{>0}$ with $a > L$. Note that the classical forward–backward method described in Algorithm 6, i.e.

$$x^{(k+1)} \in \text{prox}_{\alpha_k f_1}(x^{(k)} - \alpha_k \nabla f_0(x^{(k)}))$$

with $0 < \alpha_{min} < \alpha_k < \alpha_{max} < 1/L$ is recovered into the above general algorithm. In fact, the definition of the proximal operator, together with the boundedness from above of α_k, implies

$$f_1(x^{(k+1)}) + \nabla f_0(x^{(k)})^T(x^{(k+1)} - x^{(k)}) + \frac{1}{2\alpha_{max}}\|x^{(k+1)} - x^{(k)}\|^2 \leq f_1(x^{(k)})$$

which is condition (2.32), while the optimality condition

$$\alpha_k v^{(k+1)} + \alpha_k \nabla f_0(x^{(k)}) + x^{(k+1)} - x^{(k)} = 0$$

in combination with the boundedness from below of α_k leads to

$$\|v^{(k+1)} + \nabla f_0(x^{(k)})\| \leq \frac{1}{\alpha_{min}}\|x^{(k+1)} - x^{(k)}\|$$

namely conditions (2.33)-(2.34). Convergence of this inexact algorithm is proved in [9, Theorem 5.1]. It is clear that (2.32)-(2.34) have the role of stopping criteria for an ideal algorithm; however, no practical implementation of these criteria is provided by the authors in [9];

- inspired by the previous inexact algorithm, Chozenoux et al [47] study a variable metric forward–backward algorithm with relaxation parameters, which exploits a similar inexactness condition to (2.33)-(2.34), but evaluated at the inexact proximal point $\tilde{y}^{(k)}$ instead of the relaxed iterate $x^{(k+1)}$. Unlike the previous abstract scheme, the proximal step is computed w.r.t to a variable metric D_k which satisfies the majorization condition (2.30), while the steplength α_k is not related in any way with the Lipschitz constant of the problem, although the Lipschitz continuity of the gradient of the differentiable part is still required for convergence. The corresponding approach, denominated VMFB, is detailed in Algorithm 10. The convergence of the VMFB sequence to a stationary point is proved

Algorithm 10 inexact Variable Metric Forward-Backward (VMFB) method

```
Choose the starting point $x^{(0)} \in \mathrm{dom}(f_1)$ and fix $\tau \in \mathbb{R}_{>0}$, $\underline{\nu}, \overline{\nu} \in \mathbb{R}_{>0}$.
FOR $k = 0, 1, 2, \dots$
  STEP 1. Choose $D_k$ symmetric positive definite matrix such that,
          given $Q(x, x^{(k)}) = f_0(x^{(k)}) + \nabla f_0(x^{(k)})^T (x - x^{(k)}) + \frac{1}{2}\|x - x^{(k)}\|^2_{D_k}$, we have
              $\underline{\nu} I_n \preccurlyeq D_k \preccurlyeq \overline{\nu} I_n$
              $f_0(x) \leq Q(x, x^{(k)}), \quad \forall x \in \mathrm{dom}(f_1).$
  STEP 2. Choose $\alpha_k \in \mathbb{R}_{>0}$ and $\lambda_k \in [0, 1]$ according to (2.37)-(2.39).
  STEP 3. Compute $\tilde{y}^{(k)} \in \mathbb{R}^n$, $v^{(k)} \in \partial f_1(\tilde{y}^{(k)})$ such that
              $f_1(x^{(k)}) + \nabla f_0(x^{(k)})^T (\tilde{y}^{(k)} - x^{(k)}) + \frac{1}{\alpha_k}\|\tilde{y}^{(k)} - x^{(k)}\|^2_{D_k} \leq f_1(x^{(k)})$   (2.35)
              $\|v^{(k)} + \nabla f_0(x^{(k)})\| \leq \tau \|\tilde{y}^{(k)} - x^{(k)}\|_{D_k}.$   (2.36)
  STEP 4. Compute $x^{(k+1)} = x^{(k)} + \lambda_k(\tilde{y}^{(k)} - x^{(k)})$.
END
```

assuming that f is a KL function and that the parameters α_k and λ_k are linked by means of the following relations [47, Theorem 4.1]:

$$\underline{\eta} \leq \alpha_k \lambda_k \leq 2 - \overline{\eta} \tag{2.37}$$

$$\lambda_{min} \leq \lambda_k \leq 1 \tag{2.38}$$

$$f((1 - \lambda_k)x^{(k)} + \lambda_k y^{(k)}) \leq (1 - \underline{\alpha}) f(x^{(k)}) + \underline{\alpha} f(y^{(k)}) \tag{2.39}$$

for all $k \in \mathbb{N}$ and for some $\underline{\eta}, \overline{\eta} \in \mathbb{R}_{>0}$, $\lambda_{min} \in \mathbb{R}_{>0}$ and $\underline{\alpha} \in (0, 1]$. Similarly to what has been said for the previously described abstract algorithm, it is unclear whether and how STEP 3. and, in particular, condition (2.36), is practically ensured by some internal procedure in the numerical experience shown in [47].

Chapter 3

A novel proximal–gradient line–search based method

A certain number of issues concerning proximal–gradient methods emerges from the overview of the previous chapter. First, proximal–gradient methods often exhibit a slow rate of convergence to the solution of the optimization problem (2.1). This may be due to the fact that most approaches relate the choice of the steplength and/or relaxation parameter to the Lipschitz constant of the problem, which might be too costly to compute or lead to extremely small steplengths. Therefore, acceleration strategies, such as the adoption of a variable metric or extrapolation steps, seem unavoidable in order to turn these methods into effective tools. Second, there is a lack of convergence results in the nonconvex case, i.e. when the differentiable part f_0 in problem (2.1) is not convex. Indeed, whenever convexity is denied, the only results available for proximal–gradient methods are either the stationarity of limit points or even weaker results (see for instance Theorem 2.3). In this sense, some advances have been recently done in the literature with the introduction of the Kurdyka-Łojasiewicz assumption on the objective function; however, even in this case, most of the resulting algorithms still relate their convergence to the Lipschitz constant. Finally, classical proximal–gradient methods usually assume that a closed-form expression for the proximal operator is available whereas, in many practical cases (see Example 2.12), the proximal operator has to be computed inexactly by means of an inner iterative loop.

On the basis of these premises, we present a novel proximal–gradient algorithm, denominated VMILAn, which tackles all of the previously considered issues. First of all, VMILAn allows the possibility to adopt a variable metric in the computation of the proximal step at each iteration, that is, a scaling matrix D_k and a steplength α_k which can be both computed adaptively in an almost complete freedom, without necessarily relating them to the Lipschitz constant of the gradient. Indeed, the only requirement is that both α_k and D_k must be chosen in bounded sets. Secondly, the proximal operator in VMILAn is computed via a specific inexactness criterion, which can be practically implemented in some cases of interests for the applications. Such an inexact proximal point identifies a descent direction for the objective function. Finally, a relaxation parameter λ_k is determined along the descent direction by means of an Armijo-like rule, on which the entire convergence analysis of VMILAn relies. We would like to draw the attention on the fact that VMILAn has its roots in the Scaled Gradient Projection (SGP) method presented in Chapter 1 for differentiable optimization. Indeed, when the convex term of problem (2.1) reduces to the indicator function of a convex set and the projection operator is computed exactly, VMILAn reduces to a slightly modified version of SGP.

The following statements will be proved for the VMILAn sequences of iterates and function values:

- every limit point of the sequence, if any exists, are stationary for the objective function f; the proof of this fact is essentially based on the properties of the Armijo-like condition adopted for computing the relaxation parameter;
- if f also satisfies the Kurdyka-Łojasiewicz property, then the sequence converges to a stationary point, provided that a certain relative error condition on the subdifferential of f is satisfied at the proximal point; this condition automatically holds when the proximal point is computed exactly;
- under the same hypotheses of the previous point, it is possible to prove either finite, exponential or polynomial convergence of both the iterates and function values, according to the specific structure of the desingularizing function in the KL property.

The chapter is organized according to the following outline. In Section 3.1, a wide class of variable metric inexact descent algorithms based on an Armijo-like rule is presented. This framework also includes the proposed algorithm VMILAn, whose steps and convergence analysis is presented in section 3.2. Numerical experience on a collection of nonconvex problems is then reported in Section 3.3.

3.1 Variable metric line–search based methods

Throughout this section, the following optimization problem is addressed:

Problem 3.1. Solve

$$\min_{x\in\mathbb{R}^n} f(x) \equiv f_0(x) + f_1(x) \tag{3.1}$$

where f_0 and f_1 satisfy the following assumptions:

(i) $f_1 : \mathbb{R}^n \to \bar{\mathbb{R}}$ is proper, convex and lower semicontinuous.

(ii) $f_0 : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable on an open set $\Omega_0 \supseteq \mathrm{dom}(f_1)$.

(iii) f is bounded from below.

The proposed inexact proximal–gradient method is an instance of a more general framework developed by Bonettini et al [32] to address the nonsmooth nonconvex problem 3.1, in which the notion of proximity operator is replaced by a more general tool, in order to allow the use of non Euclidean distance in the metric.

3.1.1 A generalized forward–backward operator

Definition 3.1. *Let $\Omega \subseteq \mathbb{R}^n$ be a convex set. A family of* distance–like functions *on Ω is any set of the form $\mathcal{D}(\Omega, S) = \{d_\sigma\}_{\sigma \in S}$, where $S \subseteq \mathbb{R}^q$ is a set of parameters, $d_\sigma : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{\geq 0} \cup \{+\infty\}$ for all $\sigma \in S$ and the following conditions are satisfied for all $z, x \in \Omega$:*

($\mathcal{D}_1$) *$d_\sigma(z, x)$ is continuous in (σ, z, x);*

($\mathcal{D}_2$) *$d_\sigma(z, x)$ is continuously differentiable w.r.t. $z \in \Omega$;*

($\mathcal{D}_3$) *$d_\sigma(z, x)$ is strongly convex w.r.t. z:*

$$d_\sigma(z_2, x) \geq d_\sigma(z_1, x) + \nabla_1 d_\sigma(z_1, x)^T (z_2 - z_1) + \frac{m}{2}\|z_2 - z_1\|^2 \qquad \forall z_1, z_2 \in \Omega,$$

where $m > 0$ does not depend on σ or x (here ∇_1 denotes the gradient with respect to the first argument of a function);

($\mathcal{D}_4$) *$d_\sigma(z, x) = 0$ if and only if $z = x$ (which implies that $\nabla_1 d_\sigma(x, x) = 0$ for all $x \in \Omega$).*

Example 3.1. The above definition encompasses the following functions:

- the scaled Euclidean distance
$$d_\sigma(x, y) = \frac{1}{2\alpha}\|x - y\|_D^2 \tag{3.2}$$
with $\sigma = (\alpha, D)$, where $\alpha > 0$ and $D \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix, is an interesting example of a distance–like function in $\mathcal{D}(\mathbb{R}^n, S)$;
- the Bregman distance associated to a strongly convex function $\psi : \Omega \to \mathbb{R}$, which is defined as
$$d_\sigma(x, y) = \frac{1}{\sigma}(\psi(x) - \psi(y) - \nabla\psi(y)^T (x - y)), \quad \sigma > 0. \tag{3.3}$$

When ∇f_0 is Lipschitz continuous, a simple application of the descent lemma (Lemma 2.5) shows that, when α is sufficiently small, the following upper bound exists for f:

$$f(z) \le f(x) + \nabla f_0(x)^T(z-x) + \frac{1}{2\alpha}\|z-x\|^2 + f_1(z) - f_1(x)$$

(equality when $z = x$). In other words, a negative sign of

$$\nabla f_0(x)^T(z-x) + \frac{1}{2\alpha}\|z-x\|^2 + f_1(z) - f_1(x) \tag{3.4}$$

corresponds to a descent of the function f. Our aim now is to drop the Lipschitz assumptions on f_0 and to generalize the expression (3.4) for an arbitrary distance function d_σ replacing the squared Euclidean distance.

Definition 3.2. *Given a set of parameters $S \subseteq \mathbb{R}^q$ and $\Omega = \mathrm{dom}(f_1)$, let $\mathcal{D}(\Omega, S)$ be a set of distance-like functions and $d_\sigma \in \mathcal{D}(\Omega, S)$. The* metric function $h_\sigma : \mathbb{R}^n \times \mathbb{R}^n \to \bar{\mathbb{R}}$ *associated to d_σ is defined as*

$$h_\sigma(z,x) = \nabla f_0(x)^T(z-x) + d_\sigma(z,x) + f_1(z) - f_1(x) \quad \forall z,x \in \mathbb{R}^n. \tag{3.5}$$

Remark 3.1. We remark that h_σ depends continuously on σ, as d_σ does. Moreover, since $d_\sigma(\cdot,x)$ and f_1 are convex, proper and lower semicontinuous, $h_\sigma(\cdot,x)$ is also convex, proper and lower semicontinuous for all $x \in \Omega_0$. Finally, for any point $x \in \Omega$ and for any $d \in \mathbb{R}^n$ we have

$$h'_\sigma(x,x;d) = f'(x;d), \tag{3.6}$$

where $h'_\sigma(z,x;d)$ denotes the directional derivative of $h_\sigma(\cdot,x)$ at the point z with respect to d.

From assumption $(\mathcal{D}_3)$, it follows that $h_\sigma(\cdot,x)$ is strongly convex and, for that reason, admits a unique minimum point for any $x \in \Omega$. Hence the following definition is well-defined.

Definition 3.3. *The* generalized forward–backward operator $p : \Omega_0 \to \Omega$ *associated to any function h_σ of the form* (3.5) *is defined as*

$$p(x;h_\sigma) = \arg\min_{z\in\mathbb{R}^n} h_\sigma(z,x). \tag{3.7}$$

Remark 3.2. When d_σ is chosen as in (3.2), the operator (3.7) becomes

$$p(x;h_\sigma) = \mathrm{prox}^D_{\alpha f_1}(x - \alpha D^{-1}\nabla f_0(x)),$$

which makes $p(\cdot;h_\sigma)$ a generalization of the proximal forward–backward operator.

Under assumption $(\mathcal{D}_3)$, one can show that $p(x;h_\sigma)$ depends continuously on (x,σ).

Proposition 3.1. *Let $d_\sigma \in \mathcal{D}(\Omega, S)$ and h_σ be defined as in* (3.5). *Then $p(x;h_\sigma)$ depends continuously on (x,σ).*

Proof. Let $y = \arg\min_{z\in\mathbb{R}^n} h_\sigma(z,x)$. Then y is characterized by the equation $\nabla f_0(x) + \nabla_1 d_\sigma(y,x) + w = 0$, where $w \in \partial f_1(y)$. It follows that $f_1(u) \geq f_1(y) + w^T(u-y)$ for all $u \in \mathbb{R}^n$ or:

$$f_1(u) \geq f_1(y) - (\nabla f_0(x) + \nabla_1 d_\sigma(y,x))^T(u-y) \qquad \forall u \in \mathbb{R}^n.$$

Assumption $(\mathcal{D}_3)$ expressed in y and u gives:

$$d_\sigma(u,x) \geq d_\sigma(y,x) + \nabla_1 d_\sigma(y,x)^T(u-y) + \frac{m}{2}\|y-u\|^2 \qquad \forall u \in \mathbb{R}^n.$$

Together, these two inequalities yield:

$$\frac{m}{2}\|y-u\|^2 \leq f_1(u) - f_1(y) + d_\sigma(u,x) - d_\sigma(y,x) + \nabla f_0(x)^T(u-y) \qquad \forall u \in \mathbb{R}^n.$$

Let $y_1 = p(x_1; h_{\sigma_1})$ and $y_2 = p(x_2; h_{\sigma_2})$. Adding the previous inequality for $y = y_1$ (resp. $y = y_2$) and choosing $u = y_2$ (resp. $u = y_1$), one finds:

$$m\|y_1-y_2\|^2 \leq d_{\sigma_1}(y_2,x_1) - d_{\sigma_1}(y_1,x_1) + d_{\sigma_2}(y_1,x_2) - d_{\sigma_2}(y_2,x_2) + (\nabla f_0(x_1) - \nabla f_0(x_2))^T(y_2-y_1)$$

and hence:

$$m\|y_1-y_2\|^2 \leq d_{\sigma_2}(y_1,x_2) - d_{\sigma_1}(y_1,x_1) + d_{\sigma_1}(y_2,x_1) - d_{\sigma_2}(y_2,x_2) + \|\nabla f_0(x_1) - \nabla f_0(x_2)\|\,\|y_2-y_1\|.$$

It follows that $0 \leq \|y_1 - y_2\| \leq (b + \sqrt{b^2 + 4cm})/2m$ where $b = \|\nabla f_0(x_1) - \nabla f_0(x_2)\|$ and $c = d_{\sigma_2}(y_1,x_2) - d_{\sigma_1}(y_1,x_1) + d_{\sigma_1}(y_2,x_1) - d_{\sigma_2}(y_2,x_2)$. As f_0 is C^1, one has $\lim_{x_2\to x_1} b = 0$. As $d_\sigma(z,x)$ is continuous in (σ,z,x), one also has that $\lim_{x_2\to x_1} c = 0$. This shows then that $\lim_{x_2\to x_1}\|y_2 - y_1\| = 0$, in other words $p(x_1; h_{\sigma_1})$ is continuous in (σ_1, x_1). □

Definition 3.4. *Given a distance–like function $d_\sigma \in \mathcal{D}(\Omega, S)$ and a parameter $\gamma \in [0,1]$, the* modified metric function $\tilde{h}_{\sigma,\gamma} : \mathbb{R}^n \times \mathbb{R}^n \to \bar{\mathbb{R}}$ *is defined as*

$$\tilde{h}_{\sigma,\gamma}(z,x) = \nabla f_0(x)^T(z-x) + \gamma d_\sigma(z,x) + f_1(z) - f_1(x) \quad \forall z, x \in \mathbb{R}^n. \tag{3.8}$$

Remark 3.3. We have

$$\tilde{h}_{\sigma,\gamma}(y,x) \leq h_\sigma(y,x) \quad \forall x, y \in \mathbb{R}^n \tag{3.9}$$

and $\tilde{h}_{\sigma,\gamma} = h_\sigma$ when $\gamma = 1$.

In the following, we will show that

- Definition 2.8 of stationary point of problem 3.1 can be reformulated in terms of the fixed points of the operator $p(\cdot; h_\sigma)$, similarly to the case of the proximal forward–backward operator (see Proposition 2.11);
- the negative sign of $\tilde{h}_{\sigma,\gamma}$ detects a descent direction.

To this purpose, we collect in the following proposition some properties of the function h_σ and the associated operator $p(\cdot; h_\sigma)$.

Proposition 3.2. *Let $\sigma \in S \subseteq \mathbb{R}^q$, $\gamma \in [0,1]$, and h_σ, $\tilde{h}_{\sigma,\gamma}$ be defined as in* (3.5), (3.8), *where $d_\sigma \in \mathcal{D}(\Omega, S)$. If $x \in \Omega$ and $y = p(x; h_\sigma)$, then:*

(a) $\tilde{h}_{\sigma,\gamma}(x,x) = 0$*;*

(b) if $z \in \mathbb{R}^n$ and $\tilde{h}_{\sigma,\gamma}(z,x) < 0$, then $f'(x; z-x) < 0$;

(c) $\tilde{h}_{\sigma,\gamma}(y,x) \leq 0$ $(\tilde{h}_{\sigma,\gamma}(y,x) = 0 \Leftrightarrow y = x)$*;*

(d) $f'(x; y-x) \leq 0$ and the equality holds if and only if $\tilde{h}_{\sigma,\gamma}(y,x) = 0$ (if and only if $y = x$).

Proof. (a) is a direct consequence of definition (3.8) and condition $(\mathcal{D}_4)$ on d_σ.

(b) If $\tilde{h}_{\sigma,\gamma}(z,x) < 0$, we have

$$0 \geq -\gamma d_\sigma(z,x) > \nabla f_0(x)^T(z-x) + f_1(z) - f_1(x) \geq \nabla f_0(x)^T(z-x) + f_1'(x; z-x) = f'(x; z-x),$$

where the second inequality follows from definition (3.8) of $\tilde{h}_{\sigma,\gamma}$ and the third one from item (ii) of Remark 2.5.

(c) Since y is the minimum point of $h_\sigma(\cdot, x)$, part (a) with $\gamma = 1$ yields $h_\sigma(y,x) \leq 0$ which, in view of (3.9), gives $\tilde{h}_{\sigma,\gamma}(y,x) \leq 0$. If $y = x$, part (a) implies $\tilde{h}_{\sigma,\gamma}(y,x) = 0$. Conversely, assume $\tilde{h}_{\sigma,\gamma}(y,x) = 0$. From inequality (3.9) we have $h_\sigma(y,x) \geq 0$. On the other side, since y is the minimum point of $h_\sigma(\cdot, x)$, part (a) with $\gamma = 1$ implies $h_\sigma(y,x) \leq 0$. Thus $h_\sigma(y,x) = 0$ and since y is the unique minimizer of $h_\sigma(\cdot, x)$, we can conclude that $x = y$.

(d) From (c) we have $\tilde{h}_{\sigma,\gamma}(y,x) \leq 0$. When $\tilde{h}_{\sigma,\gamma}(y,x) < 0$ then part (b) implies $f'(x; y-x) < 0$. When $\tilde{h}_{\sigma,\gamma}(y,x) = 0$, from (c) we obtain $y = x$ and, therefore, $f'(x; y-x) = 0$. Conversely, assume $f'(x; y-x) = 0$. Using the linearity of the directional derivative and Remark 2.5, we have

$$0 = \nabla f_0(x)^T(y-x) + f_1'(x; y-x) \leq \nabla f_0(x)^T(y-x) + f_1(y) - f_1(x) \leq \tilde{h}_{\sigma,\gamma}(y,x).$$

Since $\tilde{h}_{\sigma,\gamma}(y,x) \leq 0$, we necessarily have $\tilde{h}_{\sigma,\gamma}(y,x) = 0$. □

The following proposition completely characterizes the stationary points of problem (3.1) in two equivalent ways, as fixed points of the operator $p(\cdot; h_\sigma)$, i.e. the solutions of the equation $x = p(x; h_\sigma)$, or as roots of the composite function $r_{\sigma,\gamma}(x) = \tilde{h}_{\sigma,\gamma}(p(x; h_\sigma), x)$.

Proposition 3.3. *Let $S \subseteq \mathbb{R}^q$, $\sigma \in S$, $\gamma \in [0,1]$, h_σ, $\tilde{h}_{\sigma,\gamma}$ be defined as in* (3.5) *and* (3.8)*, $x \in \Omega$ and $y = p(x; h_\sigma)$. The following statements are equivalent:*

(a) x is stationary for problem (3.1)*;*

(b) $x = y$*;*

(c) $\tilde{h}_{\sigma,\gamma}(y,x) = 0$.

Proof. (a) $\iff$ (b) Assume that $x = y$. Then, $h_\sigma(\cdot, x)$ achieves its minimum at x which, by Proposition 2.9 applied to the function $h_\sigma(\cdot, x)$, yields $h'_\sigma(x, x; z - x) \geq 0 \quad \forall z \in \mathbb{R}^n$. Recalling (3.6) we have $h'_\sigma(x, x; z - x) = f'(x; z - x)$, hence x is a stationary point for problem (3.1).

Conversely, let $x \in \Omega$ be a stationary point of problem (3.1) and assume by contradiction that $x \neq y$. Then, by Proposition 3.2 (d) we obtain $f'(x, y - x) < 0$, which contradicts the stationarity assumption on x.

(b) $\iff$ (c) See Proposition 3.2 (c). □

The knowledge that the negative sign of $\tilde{h}_{\sigma,\gamma}(y, x)$ indicates a descent direction at x is fundamental to derive the general iterative optimization algorithm of the following section.

3.1.2 A general class of line–search based algorithms

In this section we will consider a general iterative optimization algorithm which is based on the modified Armijo rule described in Algorithm LS. From now on, at each iterate $x^{(k)}$, the symbol $y^{(k)}$ will be used to indicate the minimizer of $h_{\sigma^{(k)}}(\cdot, x^{(k)})$, i.e. $y^{(k)} = p(x^{(k)}; h_{\sigma^{(k)}})$. This minimizer may be difficult to compute. We therefore introduce the symbol $\tilde{y}^{(k)}$ to indicate an approximation of $y^{(k)}$ of which, initially, we only ask $\tilde{h}_{\sigma^{(k)},\gamma}(\tilde{y}^{(k)}, x^{(k)}) < 0$. Furthermore, for the sake of simplicity, we will denote $\Omega = \mathrm{dom}(f_1)$.

Algorithm LS Modified Armijo linesearch algorithm

Let $\{x^{(k)}\}_{k\in\mathbb{N}}$, $\{\tilde{y}^{(k)}\}_{k\in\mathbb{N}}$ be two sequences of points in Ω, and $\{\sigma^{(k)}\}_{k\in\mathbb{N}}$ be a sequence of parameters in $S \subseteq \mathbb{R}^q$. Choose some $\delta, \beta \in (0, 1)$, $\gamma \in [0, 1]$. For all $k \in \mathbb{N}$ compute $\lambda^{(k)}$ as follows:

1. Set $\lambda_k = 1$ and $d^{(k)} = \tilde{y}^{(k)} - x^{(k)}$.

2. If
$$f(x^{(k)} + \lambda_k d^{(k)}) \leq f(x^{(k)}) + \beta\lambda_k \tilde{h}_{\sigma^{(k)},\gamma}(\tilde{y}^{(k)}, x^{(k)}) \tag{3.10}$$
   Then go to step 3.

   Else set $\lambda_k = \delta\lambda_k$ and go to step 2.

3. End

Algorithm LS generalizes the linesearch procedure proposed by Tseng and Yun in [140], which is recovered when d_σ is chosen as in (3.2) and $\gamma \in [0, 1)$ (the case $\gamma = 1$ is not treated in [140]). Furthermore, when $\gamma = 0$ and $f_1 = \iota_\Omega$, inequality (3.10) reduces to the classical Armijo condition (1.12) for differentiable optimization. For that reason, (3.10) may be considered as a generalization of the Armijo rule to the nondifferentiable case.

The following result guarantees that the well-posedness of Algorithm LS only depends on the negative sign of the quantity $\tilde{h}_{\sigma^{(k)},\gamma}(\tilde{y}^{(k)}, x^{(k)})$.

Proposition 3.4. *Let $\{x^{(k)}\}_{k\in\mathbb{N}}$, $\{\tilde{y}^{(k)}\}_{k\in\mathbb{N}}$ be two sequences of points in Ω, $\{\sigma^{(k)}\}_{k\in\mathbb{N}}$ a sequence of parameters in $S \subseteq \mathbb{R}^q$ and $\gamma \in [0,1]$. Assume that Ω is a closed subset of $\mathbb{R}^n$ and S a compact subset of $\mathbb{R}^q$. Then the following facts hold:*

(i) if we assume that

$$\tilde{h}_{\sigma^{(k)},\gamma}(\tilde{y}^{(k)}, x^{(k)}) < 0 \tag{3.11}$$

for all k, then Algorithm LS is well defined, i.e. for each $k \in \mathbb{N}$ the loop at step 2 terminates in a finite number of steps;

(ii) if, in addition, we assume that $\{x^{(k)}\}_{k\in\mathbb{N}}$ and $\{\tilde{y}^{(k)}\}_k \in \mathbb{N}$ are bounded sequences and $f(x^{(k+1)}) \leq f(x^{(k)})$, then we have that $\{\tilde{h}_{\sigma^{(k)},\gamma}(\tilde{y}^{(k)}, x^{(k)})\}_{k\in\mathbb{N}}$ is bounded;

(iii) assuming also that

$$\lim_{k\to\infty} f(x^{(k)}) - f(x^{(k)} + \lambda_k d^{(k)}) = 0, \tag{3.12}$$

where λ_k and $d^{(k)}$ are computed with Algorithm LS, then we have

$$\lim_{k\to\infty} \tilde{h}_{\sigma^{(k)},\gamma}(\tilde{y}^{(k)}, x^{(k)}) = 0.$$

Proof. (i) Assume by contradiction that there exists a $k \in \mathbb{N}$ such that Algorithm LS performs an infinite number of reductions, thus, for any $j \in \mathbb{N}$, we have

$$\begin{aligned}
\beta\tilde{h}_{\sigma^{(k)},\gamma}(\tilde{y}^{(k)}, x^{(k)}) &< \frac{f(x^{(k)} + \delta^j d^{(k)}) - f(x^{(k)})}{\delta^j} \\
&= \frac{f_0(x^{(k)} + \delta^j d^{(k)}) - f_0(x^{(k)})}{\delta^j} + \frac{f_1(x^{(k)} + \delta^j d^{(k)}) - f_1(x^{(k)})}{\delta^j} \\
&\leq \frac{f_0(x^{(k)} + \delta^j d^{(k)}) - f_0(x^{(k)})}{\delta^j} + \frac{\delta^j f_1(x^{(k)} + d^{(k)}) + (1-\delta^j) f_1(x^{(k)}) - f_1(x^{(k)})}{\delta^j} \\
&= \frac{f_0(x^{(k)} + \delta^j d^{(k)}) - f_0(x^{(k)})}{\delta^j} + f_1(\tilde{y}^{(k)}) - f_1(x^{(k)}),
\end{aligned}$$

where the second inequality is obtained by means of the Jensen inequality applied to the convex function f_1. Taking limits on the right hand side for $j \to \infty$ we obtain

$$\begin{aligned}
\beta\tilde{h}_{\sigma^{(k)},\gamma}(\tilde{y}^{(k)}, x^{(k)}) &\leq \nabla f_0(x^{(k)})^T d^{(k)} + f_1(\tilde{y}^{(k)}) - f_1(x^{(k)}) \\
&\leq \nabla f_0(x^{(k)})^T d^{(k)} + f_1(\tilde{y}^{(k)}) - f_1(x^{(k)}) + \gamma d_{\sigma^{(k)}}(\tilde{y}^{(k)}, x^{(k)}) \\
&= \tilde{h}_{\sigma^{(k)},\gamma}(\tilde{y}^{(k)}, x^{(k)}) < 0,
\end{aligned}$$

where the second inequality follows from the non–negativity of $d_\sigma \in \mathcal{D}(\Omega, S)$ and the last one from (3.11). Since $0 < \beta < 1$, this is an absurdum.

(ii) Assume now that $\{x^{(k)}\}_{k\in\mathbb{N}}$, $\{\tilde{y}^{(k)}\}_{k\in\mathbb{N}}$ are bounded sequences and that $f(x^{(k+1)}) \leq f(x^{(k)})$. We show that $\{\tilde{h}_{\sigma^{(k)},\gamma}(\tilde{y}^{(k)}, x^{(k)})\}_{k\in\mathbb{N}}$ is bounded. By assumption (3.11), $\tilde{h}_{\sigma^{(k)},\gamma}(\tilde{y}^{(k)}, x^{(k)})$

is bounded from above. We show that it is also bounded from below. Indeed we have

$$\begin{aligned}\tilde{h}_{\sigma^{(k)},\gamma}(\tilde{y}^{(k)},x^{(k)}) &= \nabla f_0(x^{(k)})^T(\tilde{y}^{(k)}-x^{(k)})+\gamma d_{\sigma^{(k)}}(\tilde{y}^{(k)},x^{(k)})+f_1(\tilde{y}^{(k)})-f_1(x^{(k)})\\ &\geq \nabla f_0(x^{(k)})^T(\tilde{y}^{(k)}-x^{(k)})+f_1(\tilde{y}^{(k)})-f_1(x^{(k)})\\ &= \nabla f_0(x^{(k)})^T(\tilde{y}^{(k)}-x^{(k)})+f_1(\tilde{y}^{(k)})-f(x^{(k)})+f_0(x^{(k)})\\ &\geq \nabla f_0(x^{(k)})^T(\tilde{y}^{(k)}-x^{(k)})+f_1(\tilde{y}^{(k)})-f(x^{(0)})+f_0(x^{(k)}),\end{aligned}$$

where the first inequality follows from the non–negativity of d_σ, the next line is obtained by adding and subtracting $f_0(x^{(k)})$ and the last one is a consequence of $f(x^{(k+1)})\leq f(x^{(k)})$.

As f_1 is proper and convex, there exists a supporting hyperplane, i.e. $\exists a,b\in\mathbb{R}^n$ such that $f_1(u)\geq a^Tu+b$ for all $u\in\mathbb{R}^n$. Thus:

$$\tilde{h}_{\sigma^{(k)},\gamma}(\tilde{y}^{(k)},x^{(k)})\geq\nabla f_0(x^{(k)})^T(\tilde{y}^{(k)}-x^{(k)})+a^T\tilde{y}^{(k)}+b-f(x^{(0)})+f_0(x^{(k)}).$$

The right hand side is a continuous function of $x^{(k)}$ and $\tilde{y}^{(k)}$. As these are assumed to lie on a closed and bounded set, the left hand side is bounded (from below) as well.

(iii) Let us show that the only limit point of $\{\tilde{h}_{\sigma^{(k)},\gamma}(\tilde{y}^{(k)},x^{(k)})\}_{k\in\mathbb{N}}$ is zero. To this purpose, set $\Delta^{(k)}=\tilde{h}_{\sigma^{(k)},\gamma}(\tilde{y}^{(k)},x^{(k)})$ for all $k\in\mathbb{N}$. We observe that from (3.11) and (3.12) we obtain

$$0=\lim_{k\to\infty}f(x^{(k)})-f(x^{(k)}+\lambda^{(k)}d^{(k)})=\beta\lim_{k\to\infty}\Delta^{(k)}\lambda_k. \tag{3.13}$$

Assume that there exists a subset of indices $K\subseteq\mathbb{N}$ such that $\lim_{k\in K,k\to\infty}\Delta^{(k)}=\bar{\Delta}\in\mathbb{R}$, with $\bar{\Delta}<0$. By (3.13), this implies that

$$\lim_{k\in K,k\to\infty}\lambda_k=0. \tag{3.14}$$

Denote by $\bar{K}\subseteq K$ a set of indices such that $\lim_{k\in\bar{K},k\to\infty}\sigma^{(k)}=\bar{\sigma}$, $\lim_{k\in\bar{K},k\to\infty}x^{(k)}=\bar{x}$ and $\lim_{k\in\bar{K},k\to\infty}\tilde{y}^{(k)}=\tilde{y}$ for some $\bar{\sigma}\in S$, $\bar{x},\tilde{y}\in\Omega$. From (3.14) we have that for any sufficiently large index $k\in\bar{K}$, Algorithm LS makes at least a reduction: this means that

$$\beta(\lambda_k/\delta)\Delta^{(k)}<f(x^{(k)}+(\lambda_k/\delta)d^{(k)})-f(x^{(k)}),$$

for all sufficiently large $k\in\bar{K}$. Repeating the same arguments employed in the first part of the proof, we obtain

$$\begin{aligned}\beta\Delta^{(k)} &< \frac{f_0(x^{(k)}+(\lambda_k/\delta)d^{(k)})-f_0(x^{(k)})}{\lambda_k/\delta}+f_1(\tilde{y}^{(k)})-f_1(x^{(k)})\\ &\leq \frac{f_0(x^{(k)}+(\lambda_k/\delta)d^{(k)})-f_0(x^{(k)})}{\lambda_k/\delta}+f_1(\tilde{y}^{(k)})-f_1(x^{(k)})+\gamma d_\sigma(\tilde{y}^{(k)},x^{(k)}).\end{aligned}$$

Taking limits on both sides for $k\in\bar{K},k\to\infty$, since $\{d^{(k)}=\tilde{y}^{(k)}-x^{(k)}\}_{k\in\mathbb{N}}$ is bounded and by (3.14) we obtain $\beta\bar{\Delta}\leq\bar{\Delta}<0$, which is an absurdum, being $0<\beta<1$. □

It is now possible to impose some general requirements on the sequences $\{x^{(k)}\}_{k\in\mathbb{N}}$ and $\{y^{(k)}\}_{k\in\mathbb{N}}$, in order to guarantee the global convergence of the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$.

Theorem 3.1. *Let $\{x^{(k)}\}_{k\in\mathbb{N}}$, $\{\tilde{y}^{(k)}\}_{k\in\mathbb{N}}$ be two sequences of points in Ω, $\{\sigma^{(k)}\}_{k\in\mathbb{N}} \subseteq S \subseteq \mathbb{R}^q$ and $\gamma \in [0,1]$. Assume that the following conditions hold:*

(A1) Ω is a closed subset of $\mathbb{R}^n$;

(A2) S is a compact subset of $\mathbb{R}^q$;

(A3) there exists a limit point $\bar{x}$ of $\{x^{(k)}\}_{k\in\mathbb{N}}$, with $K' \subseteq \mathbb{N}$ being a subset of indices such that $\lim_{k\in K', k\to\infty} x^{(k)} = \bar{x} \in \Omega$;

(A4) $\tilde{y}^{(k)}$ satisfies (3.11) and there exists $K'' \subseteq K'$ such that

$$\lim_{k\in K'', k\to\infty} h_{\sigma^{(k)}}(\tilde{y}^{(k)}, x^{(k)}) - h_{\sigma^{(k)}}(y^{(k)}, x^{(k)}) = 0, \quad \textit{with} \quad y^{(k)} = p(x^{(k)}; h_{\sigma^{(k)}}); \tag{3.15}$$

(A5) for any $k \in \mathbb{N}$ we have

$$f(x^{(k+1)}) \le f(x^{(k)} + \lambda_k d^{(k)}), \quad d^{(k)} = \tilde{y}^{(k)} - x^{(k)} \tag{3.16}$$

where λ_k is computed by Algorithm LS.

Then $\bar{x}$ is a stationary point for problem (3.1).

Proof. First, we notice that Algorithm LS is well defined, since (3.11) holds. We observe that, since $h_{\sigma^{(k)}}$ is strongly convex with modulus of convexity m and $y^{(k)}$ is its minimum point, we have

$$\frac{m}{2}\|z - y^{(k)}\|^2 \le h_\sigma(z, x^{(k)}) - h_{\sigma^{(k)}}(y^{(k)}, x^{(k)}) \quad \forall z \in \mathbb{R}^n. \tag{3.17}$$

Setting $z = \tilde{y}^{(k)}$ in the previous inequality and using (3.15) gives

$$\lim_{k\in K'', k\to\infty} \|\tilde{y}^{(k)} - y^{(k)}\| = 0. \tag{3.18}$$

By continuity of the operator $p(x; h_\sigma)$, since $\{x^{(k)}\}_{k\in K'}$ is bounded, $\{y^{(k)}\}_{k\in K'}$ is bounded as well. Thus, (3.18) implies that $\{\tilde{y}^{(k)}\}_{k\in K''}$ is also bounded and there exists a limit point $\bar{y}$ of $\{\tilde{y}^{(k)}\}_{k\in\mathbb{N}}$. We define $K \subseteq K''$ such that $\lim_{k\in K, k\to\infty} \tilde{y}^{(k)} = \bar{y}$ and $\lim_{k\in K, k\to\infty} \sigma^{(k)} = \bar{\sigma}$. By continuity of the operator $p(x; h_\sigma)$ with respect to all its arguments, (3.18) implies that $\bar{y} = p(\bar{x}; h_{\bar{\sigma}})$.

Consider now the sequence $\{f(x^{(k)})\}_{k\in\mathbb{N}}$. From assumption (3.16) it follows that

$$f(x^{(k+1)}) \le f(x^{(k)} + \lambda_k d^{(k)}) \le f(x^{(k)}). \tag{3.19}$$

Thus, the sequence $\{f(x^{(k)})\}_{k\in\mathbb{N}}$ is monotone nonincreasing and, therefore, it converges to some $\bar{f} \in \bar{\mathbb{R}}$. Since f is lower semicontinuous and $\bar{x}$ is a limit point of $\{x^{(k)}\}_{k\in\mathbb{N}}$, we have

$$\bar{f} = \lim_{k\to\infty} f(x^{(k)}) = \lim_{k\to\infty} f(x^{(k+1)}) \ge f(\bar{x}).$$

The previous inequality implies that $\bar{f} \in \mathbb{R}$ and this fact, together with inequality (3.19), gives

$$\lim_{k\to\infty} f(x^{(k)}) - f(x^{(k)} + \lambda_k d^{(k)}) = 0.$$

Thus we can apply Proposition 3.4 and obtain

$$\lim_{k\to\infty, k\in K} \tilde{h}_{\sigma^{(k)},\gamma}(\tilde{y}^{(k)}, x^{(k)}) = 0.$$

Combining the previous equality with (3.9) and (3.15) yields

$$0 = \lim_{k\to\infty, k\in K} \tilde{h}_{\sigma^{(k)},\gamma}(\tilde{y}^{(k)}, x^{(k)}) \le \lim_{k\to\infty, k\in K} h_{\sigma^{(k)}}(\tilde{y}^{(k)}, x^{(k)}) = \lim_{k\to\infty, k\in K} h_{\sigma^{(k)}}(y^{(k)}, x^{(k)}).$$

Since $h_{\sigma^{(k)}}(y^{(k)}, x^{(k)}) \le 0$, this implies $\lim_{k\to\infty,k\in K} h_{\sigma^{(k)}}(y^{(k)}, x^{(k)}) = 0$. Expressing inequality (3.17) for $z = x^{(k)}$, we can write

$$\frac{m}{2}\|x^{(k)} - y^{(k)}\|^2 \le h_{\sigma^{(k)}}(x^{(k)}, x^{(k)}) - h_{\sigma^{(k)}}(y^{(k)}, x^{(k)}) = -h_{\sigma^{(k)}}(y^{(k)}, x^{(k)}) \overset{k\to\infty,k\in K}{\longrightarrow} 0.$$

Thus, we proved that $\bar{y} = \bar{x}$ and, by Proposition 3.3 we have that $\bar{x}$ is stationary. □

Remark 3.4. Condition (3.11) alone is not sufficient to ensure that the limit points are stationary, but we need also to assume that (3.15) holds. As counterexample, consider the case $n = 1$, $f_0(x) = x^2/2$, $f_1(x) = 0$, $d_\sigma(x, y) = (x - y)^2/2$, $\beta = \delta = 1/2$. The sequence $x^{(k+1)} = x^{(k)} + \lambda_k(\tilde{y}^{(k)} - x^{(k)})$ with $\lambda_k = 1$, $\tilde{y}^{(k)} = x^{(k)} - (1/2)^{k+1}$ satisfies all the assumptions of Theorem 3.1 except (3.15). However, starting from $x^{(0)} = 2$, the sequence writes as $x^{(k)} = 1 + (1/2)^k \overset{k\to\infty}{\to} 1$, while the only stationary point is 0.

Conditions $(A2) - (A5)$ implicitly define a wide class of descent methods based on the Armijo condition (3.10), which are ensured to be globally convergent provided that $(A1)$ holds. The crucial ingredients of these methods are

- a descent direction $d^{(k)} = \tilde{y}^{(k)} - x^{(k)}$, where $\tilde{y}^{(k)}$ is a suitable approximation of the point $p(x^{(k)}; h_\sigma)$;
- the sufficient decrease of the objective function between two successive iterations, which has to amount at least to $\lambda_k \tilde{h}_{\sigma,\gamma}(\tilde{y}^{(k)}, x^{(k)})$, where λ_k is determined by the backtracking procedure given in Algorithm LS.

Novel algorithms may be derived from this general scheme, in which the forward–backward operator is computed *inexactly* and/or with respect to non Euclidean variable metrics.

3.2 Algorithm and convergence analysis

This section is devoted to the analysis of a novel inexact proximal–gradient algorithm, denominated *Variable Metric Inexact Line–search based Algorithm - new version (VMILAn)* [33, 111]. The caption "new version" stands for the fact that VMILAn is a modification of the method VMILA, which has been recently proposed in [32] as an instance of the general framework established in the previous section. Our aim is twofold: on one hand, we present the novel algorithm and highlight the main changes with respect to its counterpart VMILA; on the other hand, we provide a new convergence result for VMILAn under the assumption that the objective function satisfies the *Kurdyka–Łojasiewicz inequality.*

For the sake of convenience, we restate here the addressed optimization problem in which, unlike problem (3.1), the gradient of the differentiable part needs be Lipschitz continuous for the related convergence analysis.

Problem 3.2. Solve

$$\min_{x\in\mathbb{R}^n} f(x) \equiv f_0(x) + f_1(x) \tag{3.20}$$

where f_0 and f_1 satisfy the following assumptions:

(i) $f_1 : \mathbb{R}^n \to \bar{\mathbb{R}}$ is proper, convex and lower semicontinuous.

(ii) $f_0 : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable on an open set $\Omega_0 \supseteq \mathrm{dom}(f_1)$.

(iii) f_0 has an $L-$Lipschitz continuous gradient on $\mathrm{dom}(f_1)$ with $L > 0$, i.e.

$$\|\nabla f_0(x) - \nabla f_0(y)\| \leq L\|x-y\|, \ \forall \ x, y \in \mathrm{dom}(f_1).$$

(iv) f is bounded from below.

3.2.1 The proposed algorithm: VMILAn

From now on, we will assume that d_σ has the form (3.2). Therefore, given the iterate $x^{(k)} \in \mathbb{R}^n$, we will abbreviate the notation for the metric function associated to d_σ as follows

$$h_\gamma^{(k)}(y) := \tilde{h}_{\sigma,\gamma}(y, x^{(k)}) = \nabla f_0(x^{(k)})^T(x - x^{(k)}) + \frac{\gamma}{2\alpha_k}\|x - x^{(k)}\|_{D_k}^2 + f_1(x) - f_1(x^{(k)}), \tag{3.21}$$

with $\alpha_k \in \mathbb{R}_{>0}$ and D_k symmetric positive definite matrix. By also setting $h^{(k)} := h_1^{(k)}$, the proximal–gradient evaluation $y^{(k)}$ at point $x^{(k)}$ can be rewritten as

$$y^{(k)} = \arg\min_{y\in\mathbb{R}^n} h^{(k)}(y) = \mathrm{prox}_{\alpha_k f_1}^{D_k}\left(x^{(k)} - \alpha_k D_k^{-1}\nabla f_0(x^{(k)})\right). \tag{3.22}$$

Algorithm VMILAn Variable Metric Inexact Line–search based Algorithm - new version

```
Choose $0 < \alpha_{\min} \le \alpha_{\max}$, $\mu \ge 1$, $\delta, \beta \in (0,1)$, $\gamma \in [0,1]$, $\tau \in \mathbb{R}_{>0}$, $x^{(0)} \in \mathrm{dom}(f_1)$.
FOR $k = 0, 1, 2, ...$

STEP 1 Choose $\alpha_k \in [\alpha_{\min}, \alpha_{\max}]$, $D_k \in \mathcal{M}_\mu$.

STEP 2 Let $y^{(k)} = \arg\min_{y \in \mathbb{R}^n} h^{(k)}(y) = \mathrm{prox}_{\alpha_k f_1}^{D_k}\left(x^{(k)} - \alpha_k D_k^{-1} \nabla f_0(x^{(k)})\right)$.
       Compute $\tilde{y}^{(k)}$ such that
           $h^{(k)}(\tilde{y}^{(k)}) - h^{(k)}(y^{(k)}) \le -\frac{\tau}{2} h_\gamma^{(k)}(\tilde{y}^{(k)}),$    (3.23)
       where $h_\gamma^{(k)}(\tilde{y}^{(k)}) = \nabla f_0(x^{(k)})^T (\tilde{y}^{(k)} - x^{(k)}) + \frac{\gamma}{2\alpha_k} \|\tilde{y}^{(k)} - x^{(k)}\|_{D_k}^2 + f_1(\tilde{y}^{(k)}) - f_1(x^{(k)})$.

STEP 3 Set $d^{(k)} = \tilde{y}^{(k)} - x^{(k)}$.

STEP 4 Compute $\lambda_k = \delta^{i_k}$, where $i_k$ is the smallest nonnegative integer such that
           $f(x^{(k)} + \delta^{i_k} d^{(k)}) \le f(x^{(k)}) + \beta \delta^{i_k} h_\gamma^{(k)}(\tilde{y}^{(k)}).$    (3.24)

STEP 5 Compute the new point as
           $x^{(k+1)} = \begin{cases} \tilde{y}^{(k)} & \text{if } f(\tilde{y}^{(k)}) < f(x^{(k)} + \lambda_k d^{(k)}) \\ x^{(k)} + \lambda_k d^{(k)} & \text{otherwise} \end{cases}.$    (3.25)
```

Let us describe the main features of Algorithm VMILAn.

Step 1 - Variable metric.

In our approach, the steplength parameter α_k and the scaling matrix D_k should be considered as almost free parameters, which can be tuned to better capture the local features of the objective function and constraints, with the aim to accelerate the progress towards the solution. Indeed, in the following convergence analysis, we will make the only assumption that they are bounded as required at Step 1. Concerning the practical choice of the steplength parameter α_k, the general updating rules proposed in Section 1.1.2 and 1.2.2 in the context of differentiable optimization, such as the scaled Barzilai-Borwein rules (1.57) or the more recent strategies based on the Ritz values, may be applied to the differentiable part f_0 of the objective function. Unlike the steplength selection, choosing an appropriate scaling matrix D_k is strictly related to the problem features, i.e. the specific shape of the objective function to be minimized and the constraints. Some guidelines about this choice have been provided in the previous chapters, for example the Majorize-Minimize (MM) principle, based on the majorization condition (2.30) or the Split Gradient (SG) strategy, which relies upon the gradient decomposition (1.52). We refer to Section 3.3 for an extensive application of these techniques in the context of image processing.

Step 2 - Inexact computation of the proximal point

Condition (3.23) at Step 2 expresses an inexact computation of the proximal point. Let us show that such an inexactness criterion is well-posed. Since $h_\gamma^{(k)}(y) \leq h^{(k)}(y)$ for all $y \in \mathbb{R}^n$, condition (3.23) implies

$$\left(1 + \frac{\tau}{2}\right) h_\gamma^{(k)}(\tilde{y}^{(k)}) \leq h^{(k)}(y^{(k)}) \leq 0$$

where the second inequality follows from the fact that $y^{(k)}$ is the minimum point of $h^{(k)}$ and $h^{(k)}(x^{(k)}) = 0$. Hence

$$h_\gamma^{(k)}(\tilde{y}^{(k)}) \leq 0 \tag{3.26}$$

with the equality holding if and only if $\tilde{y}^{(k)}$ is stationary (Proposition 3.3). The upper bound (3.26) on $h_\gamma^{(k)}(\tilde{y}^{(k)})$ implies that $f_1(\tilde{y}^{(k)}) < +\infty$ and, since f_1 is proper, this is equivalent to say that

$$\tilde{y}^{(k)} \in \mathrm{dom}(f_1), \quad \forall\, k \in \mathbb{N}.$$

We point out that condition (3.23) is equivalent to

$$0 \in \partial_{\epsilon_k} h^{(k)}(\tilde{y}^{(k)}), \quad \text{where } \epsilon_k = -\frac{\tau}{2} h_\gamma^{(k)}(\tilde{y}^{(k)}), \tag{3.27}$$

which is a relaxed version of the inclusion characterizing the exact proximal point, i.e. $0 \in \partial h^{(k)}(y^{(k)})$. This equivalent formulation allows to compare (3.23) with other notions of inexactness recently introduced in the literature:

- (3.27) resembles the criterion proposed in [133, Definition 2.1] in the context of proximal point algorithms, in which ϵ_k is replaced by $\epsilon_k^2/(2\alpha_k)$ and no variable metric is assumed, i.e. $D_k = I_n$;

- (3.27) is weaker than the condition previously opted for VMILA in [32, Equation 31], which is

$$\frac{1}{\alpha_k} D_k(z^{(k)} - \tilde{y}^{(k)}) \in \partial_{\epsilon_k} f_1(\tilde{y}^{(k)}), \tag{3.28}$$

  where $z^{(k)} = x^{(k)} - \alpha_k D_k^{-1} \nabla f_0(x^{(k)})$ and $\{\epsilon_k\}_{k\in\mathbb{N}} \subseteq \mathbb{R}_{\geq 0}$ is either a prefixed sequence of nonnegative numbers or chosen as in (3.27). Indeed, the implication (3.28)$\Longrightarrow$(3.27) follows from

$$\left\{\frac{1}{\alpha_k} D_k(y^{(k)} - z^{(k)}) + w : w \in \partial_{\epsilon_k} f_1(y^{(k)})\right\} \subset \partial_{\epsilon_k} h^{(k)}(\tilde{y}^{(k)}),$$

  where the inclusion is strict in general (see item (vi) of Proposition 2.7).

The value ϵ_k measures the error that we make in replacing $y^{(k)}$ with $\tilde{y}^{(k)}$ at iteration k. In the next section, we will prove in Proposition 3.6 that the errors ϵ_k defined in (3.23) are summable and thus $\lim_{k\to\infty}\epsilon_k = \lim_{k\to\infty} h_\gamma^{(k)}(\tilde{y}^{(k)}) = 0$. This means that the approximate computation of the proximal point through inequality (3.23) becomes automatically more accurate as the iterations proceed.

Step 4 - Armijo-like backtracking loop
The steplength (or relaxation) parameter λ_k is adaptively computed by means of a backtracking loop at Step 4, which is the same presented in the previous section in Algorithm LS. The Armijo-like condition (3.24) accepts only steplengths which produces a sufficient decrease of the objective function and this is crucial for the convergence of the whole method. Setting $\gamma = 0$ allows to recover the standard Armijo condition and, indeed, γ can be considered as an on/off parameter to include or not the quadratic term $\|\tilde{y}^{(k)} - x^{(k)}\|^2_{D_k}$ on the right-hand-side of (3.24); in general, taking $\gamma = 1$ may produce larger steplengths (see Figure 3.1).
Thanks to (3.26), we are guaranteed that

- condition (3.24) is well-defined: in fact, since (3.26) implies that $\tilde{y}^{(k)}$ belongs to the domain of f_1, f_1 is convex and $x^{(k)} \in \mathrm{dom}(f_1)$, then any point on the line $x^{(k)} + \lambda(\tilde{y}^{(k)} - x^{(k)})$, $\lambda \in [0, 1]$ belongs to $\mathrm{dom}(f_1)$. Being $\mathrm{dom}(f_1) = \mathrm{dom}(f)$, this means that $f(x^{(k)} + \lambda d^{(k)}) < +\infty$ for all $\lambda \in [0, 1]$ and, as a consequence, the two sides of (3.24) only involve finite quantities;

- the linesearch procedure at Step 4 terminates in a finite number of steps, i.e., for all $k \in \mathbb{N}$ there exists $i_k < \infty$ such that (3.24) holds: this follows from item (i) of Proposition 3.4.

Step 5 - Overrelaxation
We observe that (3.24) does not necessarily imply that $f(x^{(k)} + \lambda_k d^{(k)}) \leq f(\tilde{y}^{(k)})$ (see Figure 3.1). This inequality is then forced to hold by Step 5, which guarantees that $f(x^{(k+1)}) \leq f(\tilde{y}^{(k)})$ and $f(x^{(k+1)}) \leq f(x^{(k)} + \lambda_k d^{(k)})$, where λ_k is computed via the backtracking loop at Step 4. This could also allow, in principle, to take a point corresponding to a smaller value of the objective function than the one obtained by simply setting $x^{(k+1)} = x^{(k)} + \lambda_k d^{(k)}$. Step 5 is the main difference between VMILA [32] and Algorithm VMILAn and it is crucial for proving the convergence of the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ in Theorem 3.3.

As a final note, we remark that VMILAn reduces to a special version of the Scaled Gradient Projection (SGP) method, presented in Algorithm 2, when $f_1 = \iota_\Omega$, $\tilde{y}^{(k)} = y^{(k)}$ and $\gamma = 0$ in Step 4. In this case, the only difference with SGP is Step 5, which can be considered as an extra step to be included in the originary version of SGP.

3.2.2 Convergence analysis

We start by collecting some properties of Algorithm VMILAn, which will be fundamental for the subsequent analysis. Here and in the following we denote by $\{x^{(k)}\}_{k\in\mathbb{N}}$, $\{\tilde{y}^{(k)}\}_{k\in\mathbb{N}}$ and $\{\lambda_k\}_{k\in\mathbb{N}}$ the sequences generated by Algorithm VMILAn.

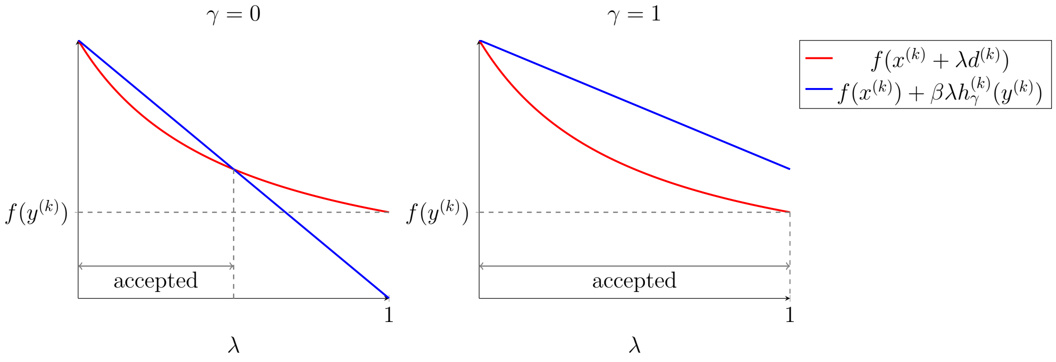


Figure 3.1: Linesearch example: $f_0(x) = \frac{2}{x+1}$, $f_1(x) = \iota_{[0,10]}(x)$, $x^{(k)} = 0$, $\beta = \frac{1}{2}$, $\alpha_k = 1$, $D_k = 1$. In general, the points satisfying the Armijo condition could not improve the function value at $y^{(k)}$.

Lemma 3.1. *For all $k \in \mathbb{N}$, the following inequality holds*

$$\frac{1}{4\alpha_{\max}\mu}\|\tilde{y}^{(k)} - x^{(k)}\|^2 \leq -(1+\tau)h_\gamma^{(k)}(\tilde{y}^{(k)}). \tag{3.29}$$

Proof. We recall that $h^{(k)}$ is $\frac{1}{\alpha_k}$ strongly convex with respect to the norm induced by D_k, i.e.

$$h^{(k)}(x) \geq h^{(k)}(y) + w^T(x-y) + \frac{1}{2\alpha_k}\|x-y\|_{D_k}^2, \quad \forall w \in \partial h^{(k)}(y). \tag{3.30}$$

Since $y^{(k)}$ is the solution of (3.22) and, thus, $0 \in \partial h^{(k)}(y^{(k)})$, from the previous inequality with $x = \tilde{y}^{(k)}$ and $y = y^{(k)}$ we have $\frac{1}{2\alpha_k}\|\tilde{y}^{(k)} - y^{(k)}\|_{D_k}^2 \leq h^{(k)}(\tilde{y}^{(k)}) - h^{(k)}(y^{(k)})$ which, in view of (3.23), gives

$$\frac{1}{2\alpha_k}\|\tilde{y}^{(k)} - y^{(k)}\|_{D_k}^2 \leq -\frac{\tau}{2}h_\gamma^{(k)}(\tilde{y}^{(k)}). \tag{3.31}$$

Exploiting again (3.30) with $x = x^{(k)}$ and $y = y^{(k)}$, recalling that $h^{(k)}(x^{(k)}) = 0$, we obtain

$$h^{(k)}(y^{(k)}) \leq -\frac{1}{2\alpha_k}\|x^{(k)} - y^{(k)}\|_{D_k}^2.$$

Combining the last inequality with (3.23) and using $h_\gamma^{(k)}(\tilde{y}^{(k)}) \leq h^{(k)}(\tilde{y}^{(k)})$ we obtain

$$\frac{1}{2\alpha_k}\|x^{(k)} - y^{(k)}\|_{D_k}^2 \leq -\left(1+\frac{\tau}{2}\right)h_\gamma^{(k)}(\tilde{y}^{(k)}).$$

By combining the triangle inequality with the previous one we obtain

$$\begin{aligned}\|x^{(k)} - \tilde{y}^{(k)}\|_{D_k} &\leq \|x^{(k)} - y^{(k)}\|_{D_k} + \|y^{(k)} - \tilde{y}^{(k)}\|_{D_k} \\ &\leq \sqrt{-(2+\tau)\alpha_k h_\gamma^{(k)}(\tilde{y}^{(k)})} + \|y^{(k)} - \tilde{y}^{(k)}\|_{D_k}\end{aligned}$$

which yields

$$\begin{aligned}\|x^{(k)}-\tilde{y}^{(k)}\|^2_{D_k} &\leq -(2+\tau)\alpha_k h^{(k)}_\gamma(\tilde{y}^{(k)})+\|y^{(k)}-\tilde{y}^{(k)}\|^2_{D_k}+\\ &\quad +2\|y^{(k)}-\tilde{y}^{(k)}\|_{D_k}\sqrt{-(2+\tau)\alpha_k h^{(k)}_\gamma(\tilde{y}^{(k)})}\\ &\leq -2(2+\tau)\alpha_k h^{(k)}_\gamma(\tilde{y}^{(k)})+2\|y^{(k)}-\tilde{y}^{(k)}\|^2_{D_k},\end{aligned}$$

where the last inequality follows from $2\sqrt{uv}\leq u+v$. Combining it with (3.31) gives

$$\begin{aligned}\frac{1}{4\alpha_k}\|x^{(k)}-\tilde{y}^{(k)}\|^2_{D_k} &\leq -(1+\frac{\tau}{2})h^{(k)}_\gamma(\tilde{y}^{(k)})+\frac{1}{2\alpha_k}\|y^{(k)}-\tilde{y}^{(k)}\|^2_{D_k}\\ &\leq -(1+\tau)h^{(k)}_\gamma(\tilde{y}^{(k)}).\end{aligned}$$

Finally, (3.29) follows from $\frac{1}{4\alpha_k}\|x^{(k)}-\tilde{y}^{(k)}\|^2_{D_k}\geq\frac{1}{4\alpha_{\max}\mu}\|x^{(k)}-\tilde{y}^{(k)}\|^2$. □

The following proposition asserts that the relaxation parameters $\{\lambda_k\}_{k\in\mathbb{N}}$ are bounded from below.

Proposition 3.5. *There exists $c\in\mathbb{R}_{>0}$ and $\lambda_{\min}\in(0,1]$ such that the following two inequalities hold:*

$$f(x^{(k)}+\lambda d^{(k)})\leq f(x^{(k)})+\lambda\left(1-cL(1+\tau)\lambda\right)h^{(k)}_\gamma(\tilde{y}^{(k)}),\quad\forall\ \lambda\in[0,1]\tag{3.32}$$

$$\lambda_k\geq\lambda_{\min},\quad\forall\ k\in\mathbb{N}.\tag{3.33}$$

Proof. In view of (3.29), setting $c=2\alpha_{\max}\mu$, one obtains

$$\|d^{(k)}\|^2\leq -2c(1+\tau)h^{(k)}_\gamma(\tilde{y}^{(k)}).\tag{3.34}$$

Since ∇f_0 is Lipschitz continuous on $\mathrm{dom}(f_1)$ with Lipschitz constant L, then from the descent lemma (Lemma 2.5) we have

$$f_0(x^{(k)}+\lambda d^{(k)})\leq f_0(x^{(k)})+\lambda\nabla f_0(x^{(k)})^T d^{(k)}+\frac{L}{2}\lambda^2\|d^{(k)}\|^2,\tag{3.35}$$

where $\lambda\in[0,1]$. By combining inequalities (3.34) and (3.35) we further obtain

$$f_0(x^{(k)}+\lambda d^{(k)})\leq f_0(x^{(k)})+\lambda\nabla f_0(x^{(k)})^T d^{(k)}-c(1+\tau)L\lambda^2 h^{(k)}_\gamma(\tilde{y}^{(k)}).$$

Summing $f_1(x^{(k)}+\lambda d^{(k)})$ on both sides of the previous relation and applying the Jensen inequality $f_1(x^{(k)}+\lambda d^{(k)})\leq(1-\lambda)f_1(x^{(k)})+\lambda f_1(\tilde{y}^{(k)})$ to the r.h.s. yields

$$\begin{aligned}f(x^{(k)}+\lambda d^{(k)}) &\leq f(x^{(k)})-\lambda f_1(x^{(k)})+\lambda f_1(\tilde{y}^{(k)})+\lambda\nabla f_0(x^{(k)})^T d^{(k)}\\ &\quad -cL\lambda^2(1+\tau)h^{(k)}_\gamma(\tilde{y}^{(k)})\\ &\leq f(x^{(k)})-\lambda f_1(x^{(k)})+\lambda f_1(\tilde{y}^{(k)})+\lambda\nabla f_0(x^{(k)})^T d^{(k)}\\ &\quad -cL\lambda^2(1+\tau)h^{(k)}_\gamma(\tilde{y}^{(k)})+\frac{\lambda\gamma}{2}\|d^{(k)}\|^2_{D_k}\\ &= f(x^{(k)})+\lambda h^{(k)}_\gamma(\tilde{y}^{(k)})-cL\lambda^2(1+\tau)h^{(k)}_\gamma(\tilde{y}^{(k)})\\ &= f(x^{(k)})+\lambda\left(1-cL(1+\tau)\lambda\right)h^{(k)}_\gamma(\tilde{y}^{(k)})\end{aligned}$$

and this proves (3.32).

The previous inequality ensures that the Armijo condition

$$f(x^{(k)} + \lambda d^{(k)}) \leq f(x^{(k)}) + \lambda\beta h_\gamma^{(k)}(\tilde{y}^{(k)}) \tag{3.36}$$

is satisfied, for all $k \in \mathbb{N}$, when $1-cL(1+\tau)\lambda \geq \beta$, that is for all λ such that $\lambda \leq (1-\beta)/(cL(1+\tau))$. If λ_k is the steplength computed by Step 5 of Algorithm VMILAn and the backtracking loop is performed at least once, then $\lambda = \lambda_k/\delta$ does not satisfy inequality (3.36), which means $\lambda_k > (1-\beta)\delta/(cL(1+\tau))$. Thus, the steplength sequence $\{\lambda_k\}_{k\in\mathbb{N}}$ satisfies inequality (3.33) with $\lambda_{\min} = (1-\beta)\delta/(cL(1+\tau))$. □

Proposition 3.6. *The sequence* $\{h_\gamma^{(k)}(\tilde{y}^{(k)})\}_{k\in\mathbb{N}}$ *is summable, i.e.*

$$0 \leq -\sum_{k=0}^{\infty} h_\gamma^{(k)}(\tilde{y}^{(k)}) < +\infty. \tag{3.37}$$

Proof. Denote by $\ell \in \mathbb{R}$ a lower bound for f, i.e. $\ell \leq f(x)\ \forall x \in \mathbb{R}^n$. From Step 5 of Algorithm VMILAn we have $f(x^{(k+1)}) \leq f(x^{(k)} + \lambda_k d^{(k)})$ and this fact, combined with (3.24) leads to

$$-\beta\lambda_k h_\gamma^{(k)}(\tilde{y}^{(k)}) \leq f(x^{(k)}) - f(x^{(k+1)}).$$

Summing the previous inequality for $k = 0, ..., j$ gives

$$-\beta\sum_{k=0}^{j} \lambda_k h_\gamma^{(k)}(\tilde{y}^{(k)}) \leq \sum_{k=0}^{j}(f(x^{(k)}) - f(x^{(k+1)})) = f(x^{(0)}) - f(x^{(j+1)}) \leq f(x^{(0)}) - \ell$$

from which we have

$$-\sum_{k=0}^{\infty} \lambda_k h_\gamma^{(k)}(\tilde{y}^{(k)}) < \infty.$$

The thesis follows by applying (3.33) to the previous series. □

Remark 3.5. An immediate consequence of Proposition 3.6 is that $\lim_{k\to+\infty} h_\gamma^{(k)}(y^{(k)}) = 0$ which, combined with (3.23), implies

$$\lim_{k\to+\infty} h^{(k)}(\tilde{y}^{(k)}) - h^{(k)}(y^{(k)}) = 0.$$

Furthermore, thanks to Step 5, we have

$$f(x^{(k+1)}) \leq f(x^{(k)} + \lambda_k d^{(k)}), \quad \forall\ k \in \mathbb{N}.$$

The two previous relations are exactly conditions $(A4)$ and $(A5)$ of Theorem 3.1. Thus, Algorithm VMILAn is a special instance of the general framework devised in Section 3.1 and, if we assumed that $\mathrm{dom}(f_1)$ is closed, we could conclude that each limit point of this sequence is stationary by means of Theorem 3.1. However, we prefer to drop the assumption of closedness of the domain and prove the global convergence of the algorithm in a more general setting.

Lemma 3.2. *The following conditions hold.*

(H1) There exists $a \in \mathbb{R}_{>0}$ such that, for all $k \in \mathbb{N}$

$$f(x^{(k+1)}) + a\|x^{(k+1)} - x^{(k)}\|^2 \leq f(x^{(k)}). \tag{H1}$$

(H2) There exists $\{\eta_k\}_{k\in\mathbb{N}} \subseteq \mathbb{R}_{\geq 0}$ such that, for all $k \in \mathbb{N}$

$$f(x^{(k+1)}) \leq f(\tilde{y}^{(k)}) \leq f(x^{(k)}) + \eta_k, \quad \lim_{k\to+\infty} \eta_k = 0. \tag{H2}$$

(H3) There exist $b \in \mathbb{R}_{>0}$, $\bar{\epsilon}_k, \hat{\epsilon}_k \in \mathbb{R}_{\geq 0}$ with $0 \leq \bar{\epsilon}_k + \hat{\epsilon}_k \leq -\frac{\tau}{2} h_\gamma^{(k)}(\tilde{y}^{(k)})$, $\zeta_k \in \mathbb{R}_{\geq 0}$, $v^{(k)} \in \{\nabla f_0(\tilde{y}^{(k)})\} + \partial_{\bar{\epsilon}_k} f_1(\tilde{y}^{(k)})$ such that, for all $k \in \mathbb{N}$

$$\|v^{(k)}\| \leq b\|x^{(k+1)} - x^{(k)}\| + \zeta_{k+1}, \quad \lim_{k\to+\infty} \zeta_k = 0. \tag{H3}$$

Proof. (H1) Combining (3.29) with the backtracking rule (3.24) immediately yields

$$f(x^{(k)} + \lambda_k d^{(k)}) \leq f(x^{(k)}) - \frac{\beta\lambda_k}{4\alpha_{\max}\mu(1+\tau)}\|\tilde{y}^{(k)} - x^{(k)}\|^2. \tag{3.38}$$

Because of (3.25), it is either $x^{(k+1)} = \tilde{y}^{(k)}$ or $x^{(k+1)} = x^{(k)} + \lambda_k d^{(k)}$. In both cases, since $\lambda_k \in [\lambda_{\min}, 1]$, we have

$$\|x^{(k+1)} - x^{(k)}\| \leq \|\tilde{y}^{(k)} - x^{(k)}\| \tag{3.39}$$

which leads to

$$f(x^{(k)} + \lambda_k d^{(k)}) \leq f(x^{(k)}) - \frac{\beta\lambda_{\min}}{4\alpha_{\max}\mu(1+\tau)}\|x^{(k+1)} - x^{(k)}\|^2.$$

Then, (H1) follows by taking $a = \frac{\beta\lambda_{\min}}{4\alpha_{\max}\mu(1+\tau)}$ and using Step 5 of Algorithm VMILAn which implies $f(x^{(k+1)}) \leq f(x^{(k)} + \lambda_k d^{(k)})$.

(H2) In order to show that (H2) holds, consider the right inequality in (3.32) with $\lambda = 1$. If $1 - cL(1+\tau) \geq 0$, then the right inequality of condition (H2) follows with $\eta_k \equiv 0$, while if $1 - cL(1+\tau) < 0$, then the inequality is satisfied by setting $\eta_k = (1 - cL(1+\tau))\, h_\gamma^{(k)}(\tilde{y}^{(k)})$ and observing that (3.37) guarantees that $\lim_{k\to\infty} \eta_k = 0$. The left inequality of (H2) follows from the definition of $x^{(k+1)}$ at Step 5 of Algorithm VMILAn.

(H3) By rewriting function $h^{(k)}$ as

$$h^{(k)}(y) = f_1(y) + \frac{1}{2\alpha_k}\|y - z^{(k)}\|^2_{D_k} - \frac{\alpha_k}{2}\|\nabla f_0(x^{(k)})\|^2_{D_k^{-1}} - f_1(x^{(k)}),$$

where $z^{(k)} = x^{(k)} - \alpha_k D_k^{-1}\nabla f_0(x^{(k)})$, we can apply item (vi) of Proposition 2.7 and equation (2.11) to compute the ϵ_k-subdifferential of $h^{(k)}$:

$$\begin{aligned}\partial_{\epsilon_k} h^{(k)}(y) &= \bigcup_{0\le\bar\epsilon_k+\hat\epsilon_k\le\epsilon_k} \partial_{\bar\epsilon_k} f_1(y) + \partial_{\hat\epsilon_k}\left(\frac{1}{2\alpha_k}\|y-z^{(k)}\|^2_{D_k}\right)\\ &= \bigcup_{0\le\bar\epsilon_k+\hat\epsilon_k\le\epsilon_k} \partial_{\bar\epsilon_k} f_1(y) + \left\{\frac{1}{\alpha_k}D_k(y-z^{(k)}+e) : \frac{\|e\|^2_{D_k}}{2\alpha_k}\le\hat\epsilon_k\right\}. \qquad (3.40)\end{aligned}$$

The point $\tilde y^{(k)}$ satisfies condition (3.23) if and only if $0 \in \partial_{\epsilon_k} h^{(k)}(\tilde y^{(k)})$, where $\epsilon_k = -\frac{\tau}{2}h_\gamma^{(k)}(\tilde y^{(k)})$. Thanks to (3.40), this ensures that there exist $\bar\epsilon_k, \hat\epsilon_k$ as above, $e^{(k)} \in \mathbb{R}^n$ satisfying $\|e^{(k)}\|_{D_k} \le \sqrt{2\alpha_k\hat\epsilon_k}$ and $w^{(k)} \in \partial_{\bar\epsilon_k} f_1(\tilde y^{(k)})$ such that

$$w^{(k)} = \frac{1}{\alpha_k}D_k(z^{(k)} - \tilde y^{(k)} + e^{(k)}). \qquad (3.41)$$

Set $v^{(k)} = \nabla f_0(\tilde y^{(k)}) + w^{(k)}$. By using the Lipschitz continuity of ∇f_0, the fact that $\alpha_k \in [\alpha_{\min}, \alpha_{\max}]$ and $D_k \in \mathcal{M}_\mu$, we have:

$$\begin{aligned}\|v^{(k)}\| &= \|\nabla f_0(\tilde y^{(k)}) + \frac{1}{\alpha_k}D_k(x^{(k)} - \alpha_k D_k^{-1}\nabla f_0(x^{(k)}) - \tilde y^{(k)} + e^{(k)})\| =\\ &= \|\nabla f_0(\tilde y^{(k)}) - \nabla f_0(x^{(k)}) + \frac{1}{\alpha_k}D_k(x^{(k)} - \tilde y^{(k)} + e^{(k)})\|\\ &\le L\|x^{(k)} - \tilde y^{(k)}\| + \frac{\mu}{\alpha_k}(\|x^{(k)} - \tilde y^{(k)}\| + \|e^{(k)}\|)\\ &\le \left(L + \frac{\mu}{\alpha_{\min}}\right)\|x^{(k)} - \tilde y^{(k)}\| + \frac{\mu}{\alpha_{\min}}\sqrt{\mu}\|e^{(k)}\|_{D_k}\\ &\le \frac{1}{\lambda_{\min}}\left(L + \frac{\mu}{\alpha_{\min}}\right)\|x^{(k+1)} - x^{(k)}\| + \left(\frac{\sqrt{2\mu^3\alpha_{\max}}}{\alpha_{\min}}\right)\sqrt{\hat\epsilon_k}.\end{aligned}$$

The thesis follows by choosing $b = \frac{1}{\lambda_{\min}}\left(L + \frac{\mu}{\alpha_{\min}}\right)$, $\zeta_k = \left(\frac{\sqrt{2\mu^3\alpha_{\max}}}{\alpha_{\min}}\right)\sqrt{\hat\epsilon_k}$ for all $k \in \mathbb{N}$ and by observing that, since

$$0 \le \zeta_k \le \left(\frac{\sqrt{2\mu^3\alpha_{\max}}}{\alpha_{\min}}\right)\sqrt{\epsilon_k} = \left(\frac{\sqrt{2\mu^3\alpha_{\max}}}{\alpha_{\min}}\right)\sqrt{-\frac{\tau}{2}h_\gamma^{(k)}(\tilde y^{(k)})}$$

and, because of (3.37), $\lim_{k\to\infty} h_\gamma^{(k)}(\tilde y^{(k)}) = 0$, then also $\lim_{k\to\infty}\zeta_k = 0$. □

Properties (H1)–(H3) are similar to conditions (C1)-(C2) proposed in [9] and discussed in Section 2.3.2. More precisely:

- (H1) coincides with (C1);

- (H2) was not contemplated in [9] and takes into account the presence of the inexact proximal point $\tilde{y}^{(k)}$ and the subsequent relaxation step;
- (H3) differs from the analogous (C2), since the vector $v^{(k)}$ here is not an exact subgradient of f and the condition also features the converging errors $\{\zeta_k\}_{k\in\mathbb{N}}$, which did not appear in [9]; a similar condition to (H3) but with exact subgradients is considered in [66].

Based on these properties, we state the following result, which claims that each limit point of the VMILAn sequence is stationary, and that the objective function f is continuous with respect to the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ and its limit points (if f_1 is continuous on its domain, the conclusion is straightforward).

Theorem 3.2. *Suppose that the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ admits a limit point $\bar{x}$. Then,*

$$\lim_{k\to\infty} f(x^{(k)}) = f(\bar{x}). \tag{3.42}$$

Moreover, $\bar{x}$ is stationary for problem (3.2)*.*

Proof. Since f is lower semicontinuous and bounded from below, and $\{f(x^{(k)})\}_{k\in\mathbb{N}}$, from (H1), is monotone nonincreasing, we have that $\lim_{k\to\infty} f(x^{(k)})$ exists and $f(\bar{x}) \leq \lim_{k\to\infty} f(x^{(k)})$. Let us show that also the opposite inequality holds. By summing inequality (H1) from $k = 0$ to N we obtain

$$a\sum_{k=0}^{N} \|x^{(k+1)} - x^{(k)}\|^2 \quad \leq \quad \sum_{k=0}^{N} f(x^{(k)}) - f(x^{(k+1)}) = f(x^{(0)}) - f(x^{(N+1)}).$$

Taking limits for $N \to \infty$ on both sides gives

$$a\sum_{k=0}^{\infty} \|x^{(k+1)} - x^{(k)}\|^2 \leq f(x^{(0)}) - f(\bar{x}) < \infty \Rightarrow \lim_{k\to\infty} \|x^{(k+1)} - x^{(k)}\| = 0. \tag{3.43}$$

Let $v^{(k)} = \nabla f_0(\tilde{y}^{(k)}) + w^{(k)}$, with $w^{(k)} \in \partial_{\bar{\epsilon}_k} f_1(\tilde{y}^{(k)})$, $\bar{\epsilon}_k \leq -\frac{\tau}{2} h_\gamma^{(k)}(\tilde{y}^{(k)})$ satisfying inequality (H3). Then, by combining (H3) and (3.43) we obtain

$$\lim_{k\to\infty} \nabla f_0(\tilde{y}^{(k)}) + w^{(k)} = \lim_{k\to\infty} v^{(k)} = 0. \tag{3.44}$$

Let $\{x^{(k_j)}\}_{j\in\mathbb{N}}$ be a subsequence of $\{x^{(k)}\}_{k\in\mathbb{N}}$ such that $\lim_{j\to\infty} x^{(k_j)} = \bar{x}$. Using Step 5 of Algorithm VMILAn and recalling that $\lambda_k \in [\lambda_{\min}, 1]$, we have

$$\lambda_{\min}^2 \|\tilde{y}^{(k)} - x^{(k)}\|^2 \leq \lambda_k^2 \|\tilde{y}^{(k)} - x^{(k)}\|^2 \leq \|x^{(k+1)} - x^{(k)}\|^2. \tag{3.45}$$

Inequality (3.45), combined with (3.43), gives $\lim_{k\to\infty} \|\tilde{y}^{(k)} - x^{(k)}\| = 0$. Then, we also have $\lim_{j\to\infty} \tilde{y}^{(k_j)} = \bar{x}$. Thus, by (3.44) and by continuity of ∇f_0, we can write

$$\lim_{j\to\infty} w^{(k_j)} = -\nabla f_0(\bar{x}). \tag{3.46}$$

Since $w^{(k_j)} \in \partial_{\bar{\epsilon}_k} f_1(\tilde{y}^{(k_j)})$, we have

$$\begin{aligned} f_1(\bar{x}) &\geq f_1(\tilde{y}^{(k_j)}) + (\bar{x} - \tilde{y}^{(k_j)})^T w^{(k_j)} - \bar{\epsilon}_{k_j} \\ &\geq f(x^{(k_j+1)}) - f_0(\tilde{y}^{(k_j)}) + (\bar{x} - \tilde{y}^{(k_j)})^T w^{(k_j)} - \bar{\epsilon}_{k_j}, \end{aligned} \tag{3.47}$$

where the second inequality follows from $f(x^{(k_j+1)}) \leq f(\tilde{y}^{(k_j)}) = f_0(\tilde{y}^{(k_j)}) + f_1(\tilde{y}^{(k_j)})$. Taking the limit of the right-hand-side for $j \to \infty$, and recalling (3.37) which implies $\lim_{j\to\infty} \bar{\epsilon}_{k_j} = 0$, we obtain

$$f_1(\bar{x}) \geq \lim_{j\to\infty} f(x^{(k_j+1)}) - f_0(\bar{x}) = \lim_{k\to\infty} f(x^{(k)}) - f_0(\bar{x})$$

which reads also as $f(\bar{x}) \geq \lim_{k\to\infty} f(x^{(k)})$ and completes the first part of the proof.
As for the second part, since $\lim_{j\to\infty} \tilde{y}^{(k_j)} = \bar{x}$, $\lim_{j\to\infty} \bar{\epsilon}_{k_j} = 0$ and (3.46) holds, we can apply Proposition 2.6 and thus obtain

$$-\nabla f_0(\bar{x}) \in \partial f_1(\bar{x}) \tag{3.48}$$

which is equivalent to $0 \in \partial f(\bar{x})$. □

We have now set the basis for our main convergence result, which will be stated in the following. The proof relies on the assumption that the objective function f satisfies the Kurdyka-Łojasiewicz (KL) property (see Definition 2.13), and is similar but not identical to Lemma 2.6 in [9] (see also [66]), since here we have to take into account of the overrelaxation at Step 5.

Theorem 3.3. *Suppose that f is a KL function and assume that the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ generated by Algorithm VMILAn satisfies the following condition*

$$\exists\ v^{(k)} \in \partial f(\tilde{y}^{(k)}) : \|v^{(k)}\| \leq b\|x^{(k+1)} - x^{(k)}\| + \zeta_{k+1}, \quad \sum_{k=1}^{\infty} \zeta_k < \infty, \tag{H4}$$

for some $b > 0$, $\zeta_k \in \mathbb{R}_{\geq 0}$, and admits a limit point $\bar{x}$. Then,

$$\sum_{k=0}^{+\infty} \|x^{(k+1)} - x^{(k)}\| < +\infty \tag{3.49}$$

and, therefore, the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ converges to $\bar{x}$, which is stationary for problem (3.2).

Proof. The stationarity of the limit points of $\{x^{(k)}\}_{k\in\mathbb{N}}$ is ensured by Proposition 3.2. It remains to show that the sequence has finite length and, thus, converges. Let υ, ϕ and U be as in Definition 2.13. These objects exist since the KL inequality holds, in particular, at $\bar{x}$. From Proposition 3.2 we have $\lim_{k\to\infty} f(x^{(k)}) = f(\bar{x})$ and, from (H2), it also follows that $\lim_{k\to\infty} f(\tilde{y}^{(k)}) = f(\bar{x})$. Consequently, the following inequality

$$f(\bar{x}) \leq f(x^{(k)}) \leq f(\tilde{y}^{(k-1)}) < f(\bar{x}) + \upsilon \tag{3.50}$$

holds for all sufficiently large k. Furthermore, let $\rho > 0$ be such that $B(\bar{x}, \rho) \subset U$. Then, using the continuity of ϕ, the fact that $\bar{x}$ is a limit point of $\{x^{(k)}\}_{k\in\mathbb{N}}$ and $\sum_k \zeta_k < \infty$, one can choose

$k_0 \in \mathbb{N}$ sufficiently large such that (3.50) holds for all $k > k_0$ and the following inequalities are satisfied:

$$\begin{aligned} \|\bar{x} - x^{(k_0)}\| \leq \frac{\rho}{4} \quad &; \quad 3\sqrt{\frac{f(x^{(k_0)}) - f(\bar{x})}{a\lambda_{\min}^2}} \leq \frac{\rho}{4} \\ \frac{b}{a}\phi(f(x^{(k_0)}) - f(\bar{x})) \leq \frac{\rho}{4} \quad &; \quad \frac{1}{b}\sum_{i=k_0+1}^{\infty} \zeta_i \leq \frac{\rho}{4}, \end{aligned}$$

a, b being the positive constants in inequalities (H1) and (H4). With a little abuse of notation, we will now use $\{x^{(k)}\}_{k\in\mathbb{N}}$ to denote the sequence $\{x^{(k+k_0)}\}_{k\in\mathbb{N}}$ and $\{\zeta_k\}_{k\in\mathbb{N}}$ instead of $\{\zeta_{k+k_0}\}_{k\in\mathbb{N}}$, so that (3.50) and the following inequality hold

$$\|\bar{x} - x^{(0)}\| + 3\sqrt{\frac{f(x^{(0)}) - f(\bar{x})}{a\lambda_{\min}^2}} + \frac{b}{a}\phi(f(x^{(0)}) - f(\bar{x})) + \frac{1}{b}\sum_{i=1}^{\infty} \zeta_i \leq \rho, \tag{3.51}$$

for all $k \geq 1$. Before we proceed with the core of the proof, let us rewrite (H1) as

$$\|x^{(k+1)} - x^{(k)}\| \leq \sqrt{\frac{f(x^{(k)}) - f(x^{(k+1)})}{a}}, \tag{3.52}$$

which, by using Step 5 of Algorithm VMILAn and (3.33), writes also as

$$\|\tilde{y}^{(k)} - x^{(k)}\| \leq \sqrt{\frac{f(x^{(k)}) - f(x^{(k+1)})}{a\lambda_{\min}^2}}. \tag{3.53}$$

Fix $k \geq 1$. We show that if $x^{(k)}, \tilde{y}^{(k-1)} \in B(\bar{x}, \rho)$, then

$$2\|x^{(k+1)} - x^{(k)}\| \leq \|x^{(k)} - x^{(k-1)}\| + \phi_k + \frac{1}{b}\zeta_k, \tag{3.54}$$

where $\phi_k = \frac{b}{a}[\phi(f(x^{(k)}) - f(\bar{x})) - \phi(f(x^{(k+1)}) - f(\bar{x}))]$. First we observe that, because of (3.50), the quantity $\phi(f(x^{(k)}) - f(\bar{x}))$ makes sense for all $k \in \mathbb{N}$, and thus ϕ_k is well defined.

If $x^{(k+1)} = x^{(k)}$, inequality (3.54) holds trivially. Then we assume $x^{(k+1)} \neq x^{(k)}$ which, thanks to (3.52), implies $f(x^{(k)}) > f(x^{(k+1)}) \geq f(\bar{x})$. Hence, from (H2) we obtain $f(\bar{x}) < f(x^{(k)}) \leq f(\tilde{y}^{(k-1)})$ which together with (3.50), gives

$$x^{(k)}, \tilde{y}^{(k-1)} \in B(\bar{x}, \rho) \cap [f(\bar{x}) < f < f(\bar{x}) + \upsilon].$$

Therefore, we can use the KL inequality in both $x^{(k)}$ and $\tilde{y}^{(k-1)}$.

Combining the KL inequality at $\tilde{y}^{(k-1)}$ with (H4) shows that $v^{(k-1)} \neq 0$ and $b\|x^{(k)} - x^{(k-1)}\| + \zeta_k \neq 0$. Since $v^{(k-1)} \in \partial f(\tilde{y}^{(k-1)})$, using again the KL inequality with (H4) we obtain

$$\phi'(f(\tilde{y}^{(k-1)}) - f(\bar{x})) \geq \frac{1}{\|v^{(k-1)}\|} \geq \frac{1}{b\|x^{(k)} - x^{(k-1)}\| + \zeta_k}. \tag{3.55}$$

Since ϕ is concave, its derivative is non increasing, thus $f(\tilde{y}^{(k-1)}) - f(\bar{x}) \geq f(x^{(k)}) - f(\bar{x})$ implies

$$\phi'(f(x^{(k)}) - f(\bar{x})) \geq \phi'(f(\tilde{y}^{(k-1)}) - f(\bar{x})).$$

Applying this fact to inequality (3.55) leads to

$$\phi'(f(x^{(k)}) - f(\bar{x})) \geq \frac{1}{b\|x^{(k)} - x^{(k-1)}\| + \zeta_k}. \tag{3.56}$$

Using the concavity of ϕ, (H1) and (3.56), we obtain

$$\begin{aligned}\phi(f(x^{(k)}) - f(\bar{x})) - \phi(f(x^{(k+1)}) - f(\bar{x})) &\geq \phi'(f(x^{(k)}) - f(\bar{x}))(f(x^{(k)}) - f(x^{(k+1)}))\\ &\geq \phi'(f(x^{(k)}) - f(\bar{x}))a\|x^{(k+1)} - x^{(k)}\|^2\\ &\geq \frac{a\|x^{(k+1)} - x^{(k)}\|^2}{b\|x^{(k)} - x^{(k-1)}\| + \zeta_k}.\end{aligned}$$

Rearranging terms in the last inequality yields

$$\|x^{(k+1)} - x^{(k)}\|^2 \leq \phi_k \left(\|x^{(k)} - x^{(k-1)}\| + \frac{1}{b}\zeta_k\right),$$

which, by applying the inequality $2\sqrt{uv} \leq u + v$, gives relation (3.54).

We are now going to establish that for $k = 1, 2, \ldots$

$$x^{(k)}, \tilde{y}^{(k-1)} \in B(\bar{x}, \rho), \tag{3.57}$$

$$\sum_{i=1}^{k} \|x^{(i+1)} - x^{(i)}\| + \|x^{(k+1)} - x^{(k)}\| \leq \|x^{(1)} - x^{(0)}\| + \chi_k + \frac{1}{b}\sum_{i=1}^{k}\zeta_i, \tag{3.58}$$

where $\chi_k = \frac{b}{a}[\phi(f(x^{(1)}) - f(\bar{x})) - \phi(f(x^{(k+1)}) - f(\bar{x}))]$.

Let us prove (3.57) and (3.58) by induction. Using (3.52) with $k = 0$ we have

$$\|x^{(1)} - x^{(0)}\| \leq \sqrt{\frac{f(x^{(0)}) - f(x^{(1)})}{a}} \leq \sqrt{\frac{f(x^{(0)}) - f(\bar{x})}{a}}. \tag{3.59}$$

Combining the above equation with (3.51) and using the triangle inequality, we obtain

$$\begin{aligned}\|\bar{x} - x^{(1)}\| &\leq \|\bar{x} - x^{(0)}\| + \|x^{(0)} - x^{(1)}\|\\ &\leq \|\bar{x} - x^{(0)}\| + \sqrt{\frac{f(x^{(0)}) - f(\bar{x})}{a}} < \rho,\end{aligned}$$

namely $x^{(1)} \in B(\bar{x}, \rho)$. Using (3.53) with $k = 0$ and applying the same arguments as before, we also have $\tilde{y}^{(0)} \in B(\bar{x}, \rho)$. Finally, direct use of (3.54) shows that (3.58) holds with $k = 1$.

By induction, suppose that (3.57) and (3.58) hold for some $k = j \geq 1$. First we prove that $x^{(j+1)} \in B(\bar{x}, \rho)$. We have

$$\begin{aligned}
\|x^{(j+1)} - \bar{x}\| &\leq \|x^{(0)} - \bar{x}\| + \|x^{(0)} - x^{(1)}\| + \sum_{i=1}^{j} \|x^{(i+1)} - x^{(i)}\| \\
&\leq \|x^{(0)} - \bar{x}\| + 2\|x^{(0)} - x^{(1)}\| + \chi_j + \frac{1}{b}\sum_{i=1}^{j} \zeta_i \\
&\leq \|x^{(0)} - \bar{x}\| + 2\sqrt{\frac{f(x^{(0)}) - f(\bar{x})}{a}} + \frac{b}{a}\phi(f(x^{(0)}) - f(\bar{x})) + \frac{1}{b}\sum_{i=1}^{j} \zeta_i \\
&< \rho,
\end{aligned}$$

where the first inequality follows from the triangle inequality, the second one from (3.58) with $k = j$, the third one from (3.59) and the monotonicity of ϕ and the last one from (3.51). Similarly, we can prove that $y^{(j)} \in B(\bar{x}, \rho)$. Noticing that $f(\bar{x}) \leq f(x^{(k+1)}) \leq f(x^{(k)}) \leq f(x^{(0)})$, (3.53) yields

$$\|\tilde{y}^{(j)} - x^{(j)}\| \leq \sqrt{\frac{f(x^{(0)}) - f(\bar{x})}{a\lambda_{\min}^2}}.$$

By using the above relation, the triangle inequality, (3.58) with $k = j$, the monotonicity of ϕ and (3.51), we have

$$\begin{aligned}
\|\bar{x} - \tilde{y}^{(j)}\| \leq\ & \|\bar{x} - x^{(0)}\| + \|x^{(0)} - x^{(1)}\| + \sum_{i=1}^{j} \|x^{(i+1)} - x^{(i)}\| + \|x^{(j+1)} - x^{(j)}\| + \|x^{(j)} - \tilde{y}^{(j)}\| \\
\leq\ & \|\bar{x} - x^{(0)}\| + 2\|x^{(0)} - x^{(1)}\| + \chi_j + \frac{1}{b}\sum_{i=1}^{j} \zeta_i + \|x^{(j)} - \tilde{y}^{(j)}\| \\
\leq\ & \|\bar{x} - x^{(0)}\| + 3\sqrt{\frac{f(x^{(0)}) - f(\bar{x})}{a\lambda_{\min}^2}} + \frac{b}{a}\phi(f(x^{(0)}) - f(\bar{x})) + \frac{1}{b}\sum_{i=1}^{j} \zeta_i \\
\leq\ & \rho,
\end{aligned}$$

or equivalently $\tilde{y}^{(j)} \in B(\bar{x}, \rho)$. Now we observe that (3.54) with $k = j + 1$ writes as

$$2\|x^{(j+2)} - x^{(j+1)}\| \leq \|x^{(j+1)} - x^{(j)}\| + \phi_{j+1} + \frac{1}{b}\zeta_{j+1}.$$

Adding the above inequality with (3.58) (with $k = j$) yields (3.58) with $k = j + 1$, which completes the induction proof.

By directly using (3.58), we get

$$\sum_{i=1}^{k} \|x^{(i+1)} - x^{(i)}\| \leq \|x^{(1)} - x^{(0)}\| + \frac{b}{a}\phi(f(x^{(1)}) - f(\bar{x})) + \frac{1}{b}\sum_{i=1}^{k} \zeta_i$$

and (on account of (H4)) therefore

$$\sum_{i=1}^{+\infty} \|x^{(i+1)} - x^{(i)}\| < +\infty,$$

which implies that the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ converges to some x^*. Considering that $\bar{x}$ is a limit point of the sequence, it must be $x^* = \bar{x}$. □

When $\tilde{y}^{(k)} = y^{(k)} = \mathrm{prox}_{\alpha_k f_1}^{D_k}(z^{(k)})$, we have $0 \in \partial h^{(k)}(y^{(k)})$ and, by remaking the same passages in Lemma 3.2 with $\epsilon_k = 0$, it follows that (H4) is automatically guaranteed with $\zeta_k \equiv 0$. When this choice is made, Algorithm VMILAn becomes an exact proximal–gradient method, whose convergence properties are stated in the following corollary, which is a direct consequence of Lemma 3.2 and Theorem 3.3.

Corollary 3.1. *Suppose that f is a KL function. Let $\{x^{(k)}\}_{k\in\mathbb{N}}$ and $\{\lambda_k\}_{k\in\mathbb{N}}$ be the sequences generated by Algorithm VMILAn with $\tilde{y}^{(k)} = y^{(k)}$ for all $k \geq 0$. If there exists a limit point $\bar{x}$ of $\{x^{(k)}\}_{k\in\mathbb{N}}$, then*

(i) $\lim_{k\to\infty} f(x^{(k)}) = f(\bar{x})$*;*

(ii) $\bar{x}$ is a stationary point for problem 3.2;

(iii) the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ converges to $\bar{x}$ and has finite length.

3.2.3 Convergence rates

We now investigate the convergence rate of Algorithm VMILAn. In particular, we follow the same outline given in [66], in which three convergence results are proved for a similar abstract descent method when the desingularizing function ϕ in Definition 2.13 is of the form $\phi(t) = \frac{C}{\theta}t^{\theta}$, with $C > 0$ and $\theta \in (0, 1]$. In Section 2.3.1, we have seen that this assumption on ϕ holds for continuous sub-analytic functions on a closed domain, real analytic functions, semi-algebraic functions and the sum of a real analytic function and a semi-algebraic function. Unlike in [66], we do not restrict to the case where $\zeta_k \equiv 0$, but we only require that the convergence of the sequence $\{\zeta_k\}_{k\in\mathbb{N}}$ is controlled by the quantity $h_{\gamma}^{(k)}(\tilde{y}^{(k)})$.

The following theorem expresses the distance of the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ to the limit in terms of the function gap and is an adaption of [66, Theorem 3].

Theorem 3.4. *Suppose that f is a KL function and that the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ satisfies* (H4) *with*

$$\zeta_k = \mathcal{O}(h_{\gamma}^{(k)}(\tilde{y}^{(k)})). \tag{3.60}$$

Assume in addition that $\{x^{(k)}\}$ admits a limit point $\bar{x}$. Let ϕ be as in Definition 2.13 for the point $\bar{x}$ and set $\bar{\phi}(t) = \max\{\phi(t), \sqrt{t}\}$. Then, there exists $M \in \mathbb{R}_{>0}$ such that

$$\|x^{(k)} - \bar{x}\| \leq \left(\frac{1}{\sqrt{a}} + \frac{M}{b} + \frac{b}{a}\right)\left(\bar{\phi}(f(x^{(k-1)}) - f(\bar{x}))\right). \tag{3.61}$$

Proof. By combining (3.24), (3.23) and (3.33), one can show that

$$
\begin{aligned}
-\frac{\tau}{2} h_\gamma^{(k-1)}(\tilde{y}^{(k-1)}) &\leq \frac{\tau}{2}\left(\frac{f(x^{(k-1)})-f(x^{(k)})}{\beta\lambda_{k-1}}\right)\\
&\leq \frac{\tau}{2\beta\lambda_{\min}}\left(f(x^{(k-1)})-f(x^{(k)})\right).
\end{aligned}
$$

From (3.60) and the above inequality, there exists $M \in \mathbb{R}_{>0}$ such that

$$
\zeta_k \leq M\left(f(x^{(k-1)})-f(x^{(k)})\right), \tag{3.62}
$$

for all $k \in \mathbb{N}$.
Let $s^{(k)} := f(x^{(k)}) - f(\bar{x}) \geq 0$. If there exists $k \in \mathbb{N}$ such that $s^{(k)} = 0$, then the algorithm terminates in a finite number of steps. Then we assume that $s^{(k)} > 0$ for all $k \in \mathbb{N}$. As previously shown in the proof of Theorem 3.3, there exists $k_0 \in \mathbb{N}$ such that (3.54) holds for all $k \geq k_0$. Summing (3.54) for $k = k_0, \ldots, N$, we get

$$
\sum_{k=k_0}^{N} \|x^{(k+1)}-x^{(k)}\| \leq \|x^{(k_0)}-x^{(k_0-1)}\| + \frac{b}{a}\phi(s^{(k_0)}) + \frac{1}{b}\sum_{k=k_0}^{N}\zeta_k. \tag{3.63}
$$

By using (3.62), summing it for $k = k_0, \ldots, N$ and observing that $f(x^{(N)}) \geq f(\bar{x})$, (3.63) yields the following inequality

$$
\sum_{k=k_0}^{N} \|x^{(k+1)}-x^{(k)}\| \leq \|x^{(k_0)}-x^{(k_0-1)}\| + \frac{b}{a}\phi(s^{(k_0)}) + \frac{M}{b}s^{(k_0-1)}. \tag{3.64}
$$

Applying the triangle inequality and passing to the limit, we obtain

$$
\begin{aligned}
\|x^{(k_0)}-\bar{x}\| \leq \sum_{k=k_0}^{\infty} \|x^{(k+1)}-x^{(k)}\| &\leq \|x^{(k_0)}-x^{(k_0-1)}\| + \frac{b}{a}\phi(s^{(k_0)}) + \frac{M}{b}s^{(k_0-1)}\\
&\leq \frac{1}{\sqrt{a}}\sqrt{f(x^{(k_0-1)})-f(x^{(k_0)})} + \frac{b}{a}\phi(s^{(k_0)}) + \frac{M}{b}s^{(k_0-1)},
\end{aligned}
$$

where the last inequality follows from (3.52). Finally, recalling that $f(x^{(k_0)}) \geq f(\bar{x})$, ϕ is an increasing function and $\{s^{(k)}\}_{k\in\mathbb{N}}$ is nonincreasing, we can write

$$
\|x^{(k_0)}-\bar{x}\| \leq \frac{1}{\sqrt{a}}\sqrt{s^{(k_0-1)}} + \frac{b}{a}\phi(s^{(k_0-1)}) + \frac{M}{b}s^{(k_0-1)}. \tag{3.65}
$$

Since $s^{(k_0-1)} \leq \sqrt{s^{(k_0-1)}}$ for a sufficiently large $k_0 \in \mathbb{N}$, we conclude that $\|x^{(k_0)}-\bar{x}\| \leq \left(\frac{1}{\sqrt{a}} + \frac{M}{b} + \frac{b}{a}\right)\bar{\phi}(s^{(k_0-1)})$. □

The next result directly follows from the previous theorem and provides explicit rates of convergence, for both the function values and the iterates.

Theorem 3.5. *Suppose that f satisfies the KL property in $\bar{x}$ (a limit point of $\{x^{(k)}\}_{k\in\mathbb{N}}$) with $\phi(t) = \frac{C}{\theta}t^{\theta}$, where $C > 0$ and $\theta \in (0,1]$, and that conditions* (H4) *and* (3.60) *hold.*

(i) If $\theta = 1$, then $\{x^{(k)}\}_{k\in\mathbb{N}}$ converges in a finite number of steps.

(ii) If $\theta \in [\frac{1}{2},1)$, then there exist $d > 0$ and $\bar{k} \in \mathbb{N}$ such that

1. $f(x^{(k)}) - f(\bar{x}) = \mathcal{O}\left(e^{-d\left(k-\bar{k}\right)}\right)$
2. $\|x^{(k)} - \bar{x}\| = \mathcal{O}\left(e^{-\frac{d}{2}\left(k-\bar{k}+1\right)}\right)$.

(iii) If $\theta \in (0,\frac{1}{2})$, then there exists $\bar{k} \in \mathbb{N}$ such that

1. $f(x^{(k)}) - f(\bar{x}) = \mathcal{O}\left(\left(k-\bar{k}\right)^{-\frac{1}{1-2\theta}}\right)$
2. $\|x^{(k)} - \bar{x}\| = \mathcal{O}\left(\left(k-\bar{k}+1\right)^{-\frac{\theta}{1-2\theta}}\right)$.

Proof. First we can assume that $s^{(k)} = f(x^{(k)}) - f(\bar{x}) > 0$ for all $k \in \mathbb{N}$, since otherwise the algorithm would terminate in a finite number of steps.
Let U be as in Definition 2.13 for the point $\bar{x}$. From Theorem 1 we know that $\{x^{(k)}\}_{k\in\mathbb{N}}$ converges to $\bar{x}$ and, because of (3.45), also $\{\tilde{y}^{(k)}\}_{k\in\mathbb{N}}$ does. Therefore there exists $\bar{k} \in \mathbb{N}$ such that

$$x^{(k+1)}, \tilde{y}^{(k)} \in U \cap [f(\bar{x}) < f < f(\bar{x}) + v]$$

for all $k \geq \bar{k}$, thus allowing to apply the KL inequality in $\tilde{y}^{(k)}$.
Let us take the squares of both sides of condition (H4), divide and multiply them by b^2 and a respectively, thus obtaining

$$\frac{a}{b^2}\|v^{(k)}\|^2 \leq a\|x^{(k+1)} - x^{(k)}\|^2 + \frac{a}{b^2}\zeta_{k+1}^2 + \frac{2a}{b}\zeta_{k+1}\|x^{(k+1)} - x^{(k)}\|.$$

By applying condition (H1) to the previous inequality, we get the following relation

$$\frac{a}{b^2}\|v^{(k)}\|^2 \leq (s^{(k)} - s^{(k+1)}) + \frac{a}{b^2}\zeta_{k+1}^2 + \frac{2\sqrt{a}}{b}\zeta_{k+1}\sqrt{s^{(k)} - s^{(k+1)}}.$$

Since $\lim_{k\to\infty}\zeta_k = 0$, it is possible to choose $\bar{k} \in \mathbb{N}$ such that $\zeta_{k+1}^2 \leq \zeta_{k+1} \leq \sqrt{\zeta_{k+1}}$ holds for all $k \geq \bar{k}$. Recalling that thanks to (3.60) there exists $M > 0$ such that $\zeta_{k+1} \leq M(s^{(k)} - s^{(k+1)})$ (see (3.62)), we obtain

$$\frac{a}{b^2}\|v^{(k)}\|^2 \leq m(s^{(k)} - s^{(k+1)}),$$

where $m = 1 + \frac{a}{b^2}M + \frac{2\sqrt{aM}}{b}$.
Set $t^{(k)} = f(\tilde{y}^{(k)}) - f(\bar{x})$. Then, by multiplying each side of the inequality by $\phi'(t^{(k)})^2$, we have

$$m\phi'(s^{(k+1)})^2(s^{(k)} - s^{(k+1)}) \geq m\phi'(t^{(k)})^2(s^{(k)} - s^{(k+1)}) \geq \frac{a}{b^2}\phi'(t^{(k)})^2\|v^{(k)}\|^2 \geq \frac{a}{b^2},$$

where the extreme left inequality has been derived using condition (H2), whereas the extreme right one has been obtained by applying the KL inequality in $\tilde{y}^{(k)}$. Therefore, we have come to the following relation

$$\phi'(s^{(k+1)})^2(s^{(k)} - s^{(k+1)}) \geq \frac{a}{mb^2}. \tag{3.66}$$

Equation (3.66) is identical to [66, Theorem 3.4, Equation 6], from which *(i)*, the rates on the function values in part 1 of *(ii)* and in part 1 of *(iii)* follow immediately, whereas the rates on the iterates contained in part 2 of *(ii)* and part 2 of *(iii)* are obtained by combining the rates on the function values and Theorem 3.4. □

Since choosing $\tilde{y}^{(k)} = y^{(k)}$ at Step 2 implies that (H4) is satisfied with $\zeta_k \equiv 0$, the convergence rates of Theorem 3.4 and 3.5 hold for the exact version of Algorithm VMILAn.

3.2.4 Practical computation of the inexact proximal point

We will now explain how to address the computation of the inexact proximal point in VMILAn. At a first glance, Step 2 of Algorithm VMILAn might seem impracticable, since it still requires the knowledge of the exact proximal point $y^{(k)}$. However, computing a point $\tilde{y}^{(k)}$ satisfying (3.23) is possible, even when $y^{(k)}$ is not known, whenever the function f_1 has the form

$$f_1(x) = g(Ax) \tag{3.67}$$

where $g : \mathbb{R}^m \to \mathbb{R}$ is a proper, convex, lower semicontinuous function with prox_g easily computable in closed form, and $A \in \mathbb{R}^{m\times n}$. In this case, the *Fenchel-Moreau-Rockafellar duality formula* [143, Corollary 2.8.5] states that, if g is continuous at Ax_0 for some $x_0 \in \mathbb{R}^n$, then

$$\min_{y\in\mathbb{R}^n} h^{(k)}(y) = \min_{y\in\mathbb{R}^n} \max_{v\in\mathbb{R}^m} F^{(k)}(y,v) = \max_{v\in\mathbb{R}^m} \Psi^{(k)}(v) \tag{3.68}$$

where $F^{(k)}$ and $\Psi^{(k)}$ are the primal–dual and dual functions associated to (3.22), respectively. The primal-dual function can be obtained from the primal one by applying the equality $g(Ax) = \max_{v\in\mathbb{R}^m} v^T Ax - g^*(v)$, which holds because of the biconjugation theorem (Theorem 2.1), to the primal function $h^{(k)}$, yielding

$$F^{(k)}(y,v) = \frac{1}{2\alpha_k}\|y - z^{(k)}\|^2_{D_k} + y^T A^T v - g^*(v) - f_1(x^{(k)}) - \frac{\alpha_k}{2}\|\nabla f_0(x^{(k)})\|^2_{D_k^{-1}} \tag{3.69}$$

with $z^{(k)} = x^{(k)} - \alpha_k D_k^{-1}\nabla f_0(x^{(k)})$. The dual function is then obtained by computing the minimum point of the primal–dual function with respect to y and replacing it into (3.69). Since $F^{(k)}(\cdot, v)$ is differentiable and convex for all $v \in \mathbb{R}^m$, the requested y is the solution of $\nabla_y F^{(k)}(y,v) = 0$, which gives $y(v) = z^{(k)} - \alpha_k D_k^{-1} A^T v$. Then the following passages lead to the explicit expression of the dual function:

$$\begin{aligned}\Psi^{(k)}(v) =&F^{(k)}(y(v),v)\\ =&\frac{1}{2\alpha_k}\|z^{(k)}-\alpha_k D_k^{-1}A^Tv-z^{(k)}\|_{D_k}^2\\ &+(z^{(k)}-\alpha_k D_k^{-1}A^Tv)^TA^Tv-g^*(v)-f_1(x^{(k)})-\frac{\alpha_k}{2}\|\nabla f_0(x^{(k)})\|_{D_k^{-1}}^2\\ =&\frac{1}{2\alpha_k}\|\alpha_k D_k^{-1}A^Tv\|_{D_k}^2+v^TAz^{(k)}\\ &-\frac{1}{\alpha_k}\|\alpha_k D_k^{-1}A^Tv\|_{D_k}^2-g^*(v)-f_1(x^{(k)})-\frac{\alpha_k}{2}\|\nabla f_0(x^{(k)})\|_{D_k^{-1}}^2\\ =&-\frac{1}{2\alpha_k}\|\alpha_k D_k^{-1}A^Tv-z^{(k)}\|_{D_k}^2-g^*(v)-f_1(x^{(k)})-\frac{\alpha_k}{2}\|\nabla f_0(x^{(k)})\|_{D_k^{-1}}^2+\frac{1}{2\alpha_k}\|z^{(k)}\|_{D_k}^2.\end{aligned}\tag{3.70}$$

By definition of the primal–dual and dual functions, the following inequalities hold

$$h^{(k)}(y)\geq F^{(k)}(y,v)\geq \Psi^{(k)}(v)\quad \forall y\in\mathbb{R}^n,\ v\in\mathbb{R}^m.\tag{3.71}$$

We now provide a sufficient condition to determine a point $\tilde{y}^{(k)}$ satisfying (3.23), which is expressed in terms of the primal and dual functions (3.68).

Lemma 3.3. *Let $h^{(k)}$, $\Psi^{(k)}$ be the primal and dual functions defined in* (3.68)*. If there exist $\tilde{y}^{(k)}\in\mathbb{R}^n$, $v^{(k)}\in\mathrm{dom}(\Psi^{(k)})$ such that*

$$h^{(k)}(\tilde{y}^{(k)})\leq \eta\Psi^{(k)}(v^{(k)}),\tag{3.72}$$

with $\eta=2/(2+\tau)$, then the point $\tilde{y}^{(k)}$ satisfies (3.23)*.*

Proof. If inequality (3.72) holds we have

$$h^{(k)}(\tilde{y}^{(k)})-h^{(k)}(y^{(k)})\leq h^{(k)}(\tilde{y}^{(k)})-\Psi^{(k)}(v^{(k)})\leq -\frac{\tau}{2}h^{(k)}(\tilde{y}^{(k)})\leq -\frac{\tau}{2}h_\gamma^{(k)}(\tilde{y}^{(k)}),\tag{3.73}$$

where the leftmost inequality follows from (3.71), while the last inequality is a consequence of $0\leq\gamma\leq 1$. □

Unlike (3.23), in condition (3.72) the dependence on the exact proximal point $y^{(k)}$ has disappeared. This allows to compute the inexact point $\tilde{y}^{(k)}$ by applying an iterative method to the dual problem (3.68), as stated in the following result.

Proposition 3.7. *Suppose that g is continuous on its domain and $\eta\in(0,1]$. For all $k\in\mathbb{N}$, let $\Psi^{(k)}$ be the dual function defined in* (3.68) *and $\{v^{(k,\ell)}\}_{\ell\in\mathbb{N}}\subseteq\mathrm{dom}(\Psi^{(k)})$ a sequence such that*

$$\lim_{\ell\to+\infty} v^{(k,\ell)}=\operatorname*{argmax}_{v\in\mathbb{R}^m}\Psi^{(k)}(v)\tag{3.74}$$

$$\lim_{\ell\to+\infty}\Psi^{(k)}(v^{(k,\ell)})=\max_{v\in\mathbb{R}^m}\Psi^{(k)}(v).\tag{3.75}$$

Set $\tilde{y}^{(k,\ell)} = z^{(k)} - \alpha_k D_k^{-1} A^T v^{(k,\ell)}$. *Then there exists* $\tilde{\ell} \in \mathbb{N}$ *such that*

$$h^{(k)}(\tilde{y}^{(k,\ell)}) \leq \eta \Psi^{(k)}(v^{(k,\ell)}), \quad \forall\ \ell \geq \tilde{\ell}. \tag{3.76}$$

Proof. Set $a_\ell = \Psi^{(k)}(v^{(k,\ell)})$. Since inequalities (3.71) are satisfied, in particular, for $y = y^{(k)}$, it follows that $a_\ell \leq h^{(k)}(y^{(k)})$ and, by hypothesis, it also holds $\lim_\ell a_\ell = \max_v \Psi^{(k)}(v) = h^{(k)}(y^{(k)})$. Combining these two facts and noting that $\eta \in (0,1]$, we have

$$h^{(k)}(y^{(k)}) < \eta a_\ell \tag{3.77}$$

for all sufficiently large ℓ. We observe that

$$\begin{aligned} y^{(k)} = \underset{y}{\operatorname{argmin}}\, h^{(k)}(y) &\iff \frac{1}{\alpha_k} D_k(z^{(k)} - y^{(k)}) \in A^T \partial g(Ay^{(k)}) \\ &\iff y^{(k)} = z^{(k)} - \alpha_k D_k^{-1} A^T v \end{aligned}$$

where

$$v \in \partial g(Ay^{(k)}) \iff Ay^{(k)} \in \partial g^*(v) \iff 0 \in \partial \Psi^{(k)}(v).$$

Hence $v = v_* = \operatorname{argmax}_{v \in \mathbb{R}^m} \Psi^{(k)}(v)$ and $y^{(k)} = z^{(k)} - \alpha_k D_k^{-1} A^T v_*$. Condition (3.74) guarantees that

$$\lim_{\ell \to +\infty} \tilde{y}^{(k,\ell)} = y^{(k)}.$$

Since g is continuous, so does $h^{(k)}$, and therefore

$$\lim_{\ell \to +\infty} h^{(k)}(\tilde{y}^{(k,\ell)}) = h^{(k)}(y^{(k)}).$$

The above limit, in combination with (3.77) and the inequality $h^{(k)}(y^{(k)}) \leq h^{(k)}(\tilde{y}^{(k,\ell)})$, allows to conclude that (3.76) is eventually satisfied for ℓ sufficiently large. □

Remark 3.6. Since $v^{(k,\ell)} \in \operatorname{dom}(\Psi^{(k)})$ for all $\ell \in \mathbb{N}$, any point $\tilde{y}^{(k,\ell)}$ satisfying (3.76) belongs to $\operatorname{dom}(f_1)$. However, we are not guaranteed that this is the case for any point of the primal sequence, i.e. that $\tilde{y}^{(k,\ell)} \in \operatorname{dom}(f_1)$ for all $\ell \in \mathbb{N}$. Therefore, a practical issue arises in implementing (3.76), since such a condition might involve infinite quantities in the process.
If $\operatorname{dom}(f_1)$ is closed, this issue may be fixed by considering the sequence $\bar{y}^{(k,\ell)} = P_{\operatorname{dom}(f_1)}(\tilde{y}^{(k,\ell)})$, where $P_{\operatorname{dom}(f_1)}$ denotes the Euclidean projection onto $\operatorname{dom}(f_1)$, and stopping the iterative procedure when the inequality

$$h^{(k)}(\bar{y}^{(k,\ell)}) \leq \eta \Psi^{(k)}(v^{(k,\ell)}) \tag{3.78}$$

is satisfied. This condition is well-posed, since the continuity of $P_{\operatorname{dom}(f_1)}$ and the fact that $y^{(k)} \in \operatorname{dom}(f_1)$ guarantee again that $\bar{y}^{(k,\ell)} \to y^{(k)}$ and thus that the procedure terminates in a finite number of steps.

In a nutshell, an inexact proximal point $\tilde{y}^{(k)}$ satisfying (3.23) can be computed by implementing the following steps:

1. generate a sequence $\{v^{(k,\ell)}\}_{\ell\in\mathbb{N}}$ where the iterates and function values converge to the solution of the dual problem (3.68) and its optimal value, respectively. For instance, an algorithm complying with these two properties is FISTA in the variant proposed by Chambolle and Dossal [42] (see Algorithm 9);

2. compute the sequence $\{\bar{y}^{(k,\ell)}\}_{\ell\in\mathbb{N}}$ where $\bar{y}^{(k,\ell)} = P_{\mathrm{dom}(f_1)}(z^{(k)} - \alpha_k D_k^{-1} A^T v^{(k,\ell)})$;

3. set $\tilde{y}^{(k)} = \bar{y}^{(k,\ell)}$, where ℓ is the first nonnegative integer such that the stopping criterion (3.78) is satisfied.

Then, at each iteration of VMILAn, an inner loop is required in order to compute $\tilde{y}^{(k)}$. Clearly, the entire procedure depends on the value of the parameter η, which measures the quality of the inexact proximal point we are computing: the closer η is to 1, the more precise the approximation gets, but at the cost of a larger number of inner iterations, and viceversa when η is approaching 0. However, numerical experience shows that a good balance between convergence speed and computational cost in VMILAn can be achieved, as remarked in [32].

Finally, let us observe that (3.67) includes also the case where $f_1(x)$ is defined as

$$f_1(x) = \sum_{i=1}^{r} g_i(A_i x),$$

where $A_i \in \mathbb{R}^{m_i\times n}$, $g_i : \mathbb{R}^{m_i} \to \mathbb{R}$. Indeed, formulation (3.67) is recovered by setting $A = [A_1^T\ A_2^T\ ...\ A_r^T]^T \in \mathbb{R}^{m\times n}$ with $m = \sum_{i=1}^r m_i$ and $g(t) = \sum_{i=1}^r g_i(t_i)$ for all $t = (t_1, \dots, t_r) \in \mathbb{R}^m$, with $t_i \in \mathbb{R}^{m_i}$. In this case the dual variable v can be partitioned as $v = [v_1^T\ v_2^T ...\ v_r^T]^T$, where $v_i \in \mathbb{R}^{m_i}$ and $g^*(v) = \sum_{i=1}^r g_i^*(v_i)$ (see Proposition 2.3).

3.3 Applications in image processing

In order to confirm the efficiency of the proposed algorithm, we carry out different numerical experiments on realistic nonconvex optimization problems arising in image processing. We compare the obtained results with those provided by some recent methods already applied in such a framework. All the numerical results in the following sections have been obtained on a PC equipped with an INTEL Core i7 processor 2.70GHz with 8GB of RAM running Matlab ver 7 R2010b.

3.3.1 Image deconvolution in presence of signal dependent Gaussian noise

In this section we consider the image restoration problem described in [47], where the observed data $g \in \mathbb{R}^n$ are assumed to be acquired according to the model

$$g_i = (Hx_{\text{true}})_i + \sigma_i((Hx_{\text{true}})_i)w_i,$$

where $x_{\text{true}} \in \mathbb{R}^n$ denotes the original image to be reconstructed, $H \in \mathbb{R}^{n\times n}$ is a matrix with non-negative entries representing the acquisition system, $w = (w_1, \cdots, w_n)^T$ is a realization of Gaussian random vector with zero mean and covariance matrix I_n and $\sigma_i : \mathbb{R} \to \mathbb{R}_{>0}$ is defined as

$$\sigma_i(u) = \sqrt{a_i u + b_i},$$

with $a_i \in \mathbb{R}_{\geq 0}$, $b_i \in \mathbb{R}_{>0}$, for all $i = 1, ..., n$.

The problem of recovering the unknown x_{true} from the knowledge of g can be addressed by means of the Bayesian paradigm, for which an overview can be found by the reader in Appendix A. According to this approach, an estimate of the true image x_{true} can be computed by solving the minimization problem (3.2), where f_0 is a data discrepancy function corresponding to the negative log–likelihood of the data, and f_1 is a regularization term chosen to induce some desired properties on the computed solution.

In this case, the negative log-likelihood function is given by

$$f_0(x) = \frac{1}{2}\sum_{i=1}^{n} \frac{((Hx)_i - g_i)^2}{a_i(Hx)_i + b_i} + \log(a_i(Hx)_i + b_i), \tag{3.79}$$

which is continuously differentiable and nonconvex on $\text{dom}(f_0) = \{x \in \mathbb{R}^n : a_i(Hx)_i + b_i > 0 \ \forall i = 1, ..., n\}$.

If one wants to preserve the edges in the reconstruction and also the non-negativity of the pixel values, the regularization term can be chosen as the sum of the total variation functional (see [132] or Appendix A) and the indicator function of the set $\mathbb{R}^n_{\geq 0}$, i.e.

$$f_1(x) = \rho \sum_{i=1}^{n} \|\nabla_i x\| + \iota_{\mathbb{R}^n_{\geq 0}}(x), \tag{3.80}$$

where $\rho \in \mathbb{R}_{>0}$ is a regularization parameter and $\nabla_i \in \mathbb{R}^{2\times n}$ represents the discrete gradient of the two dimensional object x at pixel i.

We remark that the assumptions of problem (3.2) are satisfied for f_0 and f_1. Indeed, f_1 is convex and continuous on $\text{dom}(f_1)$ and, since $b_i > 0$ for all $i = 1, \ldots, n$ and H has non–negative entries, we have $\text{dom}(f_0) \supset \text{dom}(f_1)$. Furthermore, ∇f_0 is Lipschitz continuous on $\text{dom}(f_1)$, in fact f_0 can be expressed as

$$f_0(x) = \sum_{i=1}^{n} \nu_1^{(i)}((Hx)_i) + \nu_2^{(i)}((Hx)_i),$$

where

$$\nu_1^{(i)}(u) = \frac{1}{2}\frac{(u-g_i)^2}{a_i u + b_i}, \quad \nu_2^{(i)}(u) = \frac{1}{2}\log(a_i u + b_i)$$

have bounded second derivatives on $\mathrm{dom}(f_0)$. Finally observe that, since f is given by sums, products and compositions of analytic functions on $\mathrm{dom}(f_0)$, then f itself is analytic [88] and, as we have seen in Section 2.3.1, this is sufficient to conclude that f is a KL function.

In order to validate the effectiveness of the proposed method, we consider the test problem "jetplane", which can be downloaded from [127] (see Figure 3.2). Here, the operator H corresponds to a convolution with a truncated Gaussian function of size 7 × 7, $a_i = b_i = 1$ for all $i = 1, ..., n$ and $\rho = 0.03$.

Algorithm VMILAn has been implemented in Matlab environment with the following settings:

Step 1 - metric selection: we consider three different choices for D_k, all leading to a diagonal matrix whose entries are defined as follows:

MM $(D_k)^{-1}_{ii} = \max\{\min\{(A_k)_{ii}, \mu\}, \frac{1}{\mu}\}$, where A_k is defined in [47, formula (36)] with $\varepsilon = 0$. This matrix A_k is introduced in [47], following the Majorization-Minimization (MM) approach, where the authors show that the quadratic function $Q(x, x^{(k)}) = f_0(x^{(k)}) + \nabla f_0(x^{(k)})^T(x - x^{(k)}) + \frac{1}{2}\|x - x^{(k)}\|^2_{A_k}$ is a majorant function for f_0, i.e. $f_0(x) \leq Q(x, x^{(k)})$ for all $x \in \mathrm{dom}(f_1)$.

SG $(D_k)^{-1}_{ii} = \max\{\min\{\frac{x_i^{(k)}}{V_i(x^{(k)})+\epsilon}, \mu\}, \frac{1}{\mu}\}$, where ϵ is set to the machine precision and $V(x^{(k)})$ is determined on the basis of the Split Gradient (SG) idea [89], i.e. in such a way that

$$\nabla f_0(x^{(k)}) = V(x^{(k)}) - U(x^{(k)}),$$

where $V(x^{(k)})$ is a vector with positive entries and $U(x^{(k)})$ has nonnegative entries. In this case, a feasible choice for the positive part is $V(x^{(k)}) = H^T s^{(k)}$ with

$$s_i^{(k)} = (Hx)_i \frac{a_i((Hx)_i + g_i) + 2b_i}{2(a_i(Hx)_i + b_i)^2} + \frac{a_i}{2(a_i(Hx)_i + b_i)}.$$

I $D_k = I_n$.

The parameter μ bounding the diagonal entries of D_k is set to 10^{10}.

Step 1 - steplength selection: once computed the matrix D_k, the stepsize parameter α_k is chosen using the adaptive strategy proposed in [108] and based on the approximation of the eigenvalues of the Hessian matrix of the objective function by means of a Lanczos–like process (see also Section 1.2.2). The idea is to apply this rule to the differentiable part f_0. In our problem, for a fixed positive integer m (in our experiments we consider $m = 3$), one has to:

a) Define the matrices

$$\widetilde{G} = \left[D_{k-m}^{-1/2}\widetilde{g}^{(k-m)}, \ldots, D_{k-1}^{-1/2}\widetilde{g}^{(k-1)}\right] \ , \ \Gamma = \begin{bmatrix} \alpha_{k-m}^{-1} & & \\ -\alpha_{k-m}^{-1} & \ddots & \\ & \ddots & \alpha_{k-1}^{-1} \\ & & -\alpha_{k-1}^{-1} \end{bmatrix},$$

by collecting m consecutive steplengths and reduced gradients

$$\widetilde{g}_j^{(k)} = \begin{cases} 0 & \text{if } x_j^{(k)} = 0, \\ \left[\nabla f_0(x^{(k)})\right]_j & \text{if } x_j^{(k)} > 0 \end{cases}. \tag{3.81}$$

b) Compute the Cholesky factorization $R^T R$ of the $m \times m$ matrix $\widetilde{G}^T\widetilde{G}$, the solution r of the linear system $R^T r = \widetilde{G}^T D_k^{-1/2}\widetilde{g}^{(k)}$ and the $m \times m$ matrix $\Phi = [R \quad r]\Gamma R^{-1}$.

c) Compute the eigenvalues of the symmetric and tridiagonal approximation $\widetilde{\Phi}$ of Φ defined as

$$\widetilde{\Phi} = \operatorname{diag}(\Phi) + \operatorname{tril}(\Phi, -1) + \operatorname{tril}(\Phi, -1)^T,$$

being $\operatorname{diag}(\cdot)$ and $\operatorname{tril}(\cdot, -1)$ the diagonal and the strictly lower triangular parts of a matrix, and use the reciprocal of the positive eigenvalues obtained as steplengths for the next iterations.

Finally, the steplengths α_k are constrained in the interval $[\alpha_{\min}, \alpha_{\max}]$, where $\alpha_{\min} = 10^{-5}$, $\alpha_{\max} = 10^2$.

Step 2 - inexact computation of the proximal point: since the proximal operator of f_1 is not available in a closed form, it has to be approximated via an iterative solution. We observe that the nonsmooth regularization term has the form $f_1(x) = g(Ax)$ where $A^T = (\nabla_1^T, \ldots, \nabla_n^T, I_n) \in \mathbb{R}^{n \times 3n}$ and $g : \mathbb{R}^{3n} \to \bar{\mathbb{R}}$ is defined as

$$g(t) = \sum_{i=1}^{n} \left\| \begin{pmatrix} t_{2i-1} \\ t_{2i} \end{pmatrix} \right\| + \iota_{\mathbb{R}^n_{\geq 0}} \begin{pmatrix} t_{2n+1} \\ \vdots \\ t_{3n} \end{pmatrix}.$$

We implement an inexact version of Algorithm VMILAn, where the approximate proximal point $\tilde{y}^{(k)}$ satisfying (3.23) is computed as described in Section 3.2.4 with $\eta = 10^{-6}$ or, equivalently, with $\tau = 2(10^6 - 1)$. In this case, in virtue of Example 2.1-2.2 and Proposition 2.3, we have that g^* is the indicator function of the set $\mathcal{C} = B^2(0, \rho) \times \cdots \times B^2(0, \rho) \times \mathbb{R}^n_{\leq 0}$, where $B^2(0, \rho)$ is the 2-dimensional ball with center 0 and radius ρ. Thus the dual problem (3.68) can be written as the following constrained least squares problem

$$\max_{v \in \mathcal{C}} -\frac{1}{2\alpha_k}\|\alpha_k D_k^{-1} A^T v - z^{(k)}\|^2_{D_k} - f_1(x^{(k)}) - \frac{\alpha_k}{2}\|\nabla f_0(x^{(k)})\|^2_{D_k^{-1}} + \frac{1}{2\alpha_k}\|z^{(k)}\|^2_{D_k}. \tag{3.82}$$

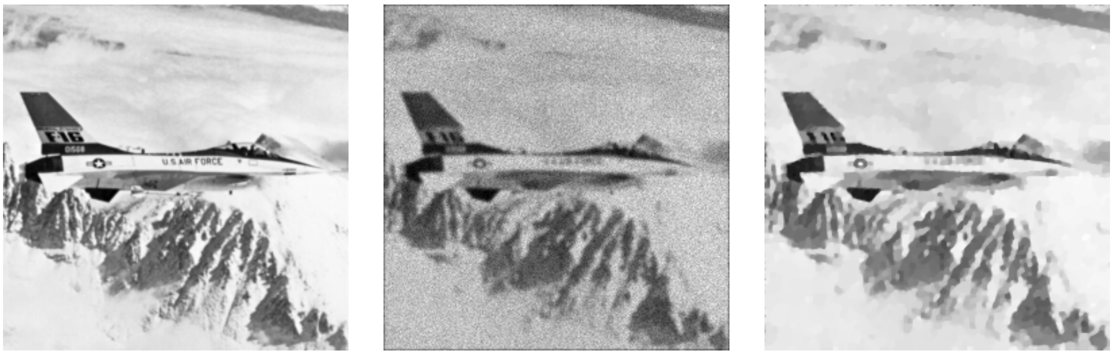


Figure 3.2: Jetplane test problem: original object (left), blurred and noisy image (middle), and VMILAn reconstruction (right).

As inner solver for the subproblem (3.82), we adopt algorithm FISTA in the variant proposed in [42] and reported in Algorithm 9 of Section 2.2.3, where we set $a = 2.1$. We remark that, if condition (H4) is not ensured on the point $\tilde{y}^{(k)}$, we could not invoke Theorem 3.3 to guarantee the convergence of the whole sequence. However, the stationarity of the limit points is guaranteed by Proposition 3.2, which holds independently of (H4).

Other parameters setting: the line–search parameters have been chosen as $\delta = 0.5$, $\beta = 10^{-4}$, $\gamma = 1$. These are standard choices for the Armijo parameters in the constrained optimization framework [37, 113, 112], where it has been remarked that the performance of the Armijo line–search is usually not sensitive to the choice of these parameters.

We compare the performances of our method with the variable metric forward backward (VMFB) algorithm [47] (see Algorithm 10 of Section 2.2.3) in the implementation provided by the authors which can be downloaded from [127]. We observed that both methods achieve the same value of the objective function in the limit, denoted by f^*, which is in general not guaranteed for nonconvex problems. Thus in this case we can compare the optimization properties of the algorithms by measuring the progress toward this value, which has been numerically approximated first by running 5000 iterations of all methods and retaining the smallest value.

Figure 3.3 reports the relative decrease of the objective function with respect to the minimum value f^* as a function of the iteration number and of the computational time. We can observe a faster decrease of the objective function for Algorithm VMILAn. The best performances are achieved by choosing $D_k = I_n$ which means that, for this specific application, the most significant benefits in Algorithm VMILAn come from the variable choice of the steplength α_k. The inner solver for computing an approximation of the proximal point requires about 2–3 iterations per outer iteration, except for the choice SG of the matrix D_k. In all experiments the first option in (3.25) never occurred. The reconstructed image obtained with VMILAn is shown in the right panel of figure 3.2.

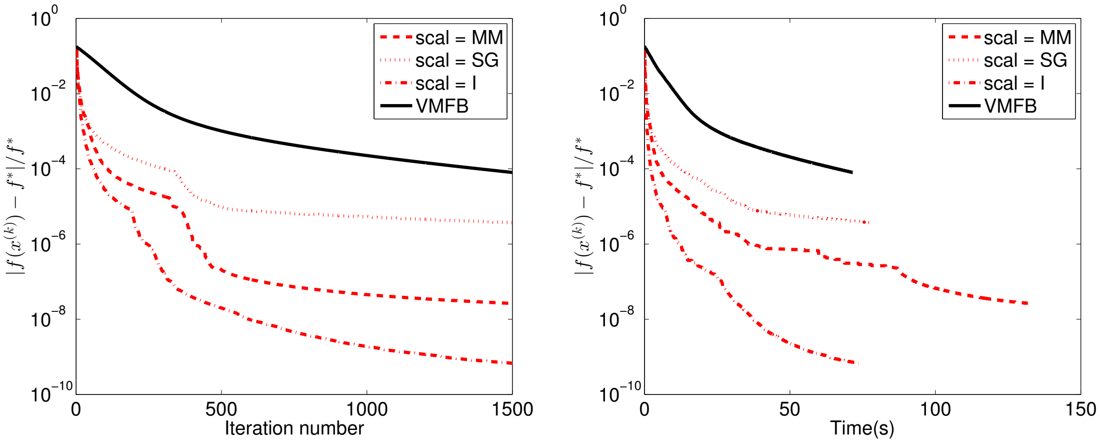


Figure 3.3: Image deconvolution in presence of signal dependent Gaussian noise. Relative decrease of the objective function toward the minimum value with respect to the iteration number (left) and computational time in seconds (right).

3.3.2 Image deblurring in presence of Cauchy noise

As a second test, we take into account the problem of recovering a blurred image corrupted by Cauchy noise. In [134] the authors propose a novel variational model aimed at facing Cauchy noise image restoration based on total variation regularization. More in detail, they suppose the degraded image $g \in \mathbb{R}^n$ can be written as $g = Hx + v$, where $x \in \mathbb{R}^n$ is the true object, $H \in \mathbb{R}^{n\times n}$ is the discretization of the blurring operator and $v \in \mathbb{R}^n$ represents the random noise which is modelled by a Cauchy probability distribution corresponding to a density of the form

$$f(v) = \frac{1}{\pi}\frac{\gamma}{\gamma^2 + v^2}, \qquad \gamma > 0.$$

The discrete version of the optimization problem they suggest can be formulated as follows

$$\min_{x\in\mathbb{R}^n} \frac{\lambda}{2}\sum_{i=1}^{n} \log\big(\gamma^2 + ((Hx)_i - g_i)^2\big) + \sum_{i=1}^{n} \|\nabla_i x\|, \tag{3.83}$$

where λ is the regularization parameter. We decide to force the solution of being nonnegative and therefore we add to the objective function in (3.83) the indicator function of the nonnegative orthant. In these settings, the nondifferentiable part of the function to minimize becomes as in (3.80) (with $\rho = 1$), while f_0 reduces to the logarithmic discrepancy. The corresponding optimization problem fits into the framework of problem (3.2). Indeed, f_0 is a real analytic function on the entire space $\mathbb{R}^n$ and thus $\mathrm{dom}(f_0) \supset \mathrm{dom}(f_1)$. Furthermore, by writing the

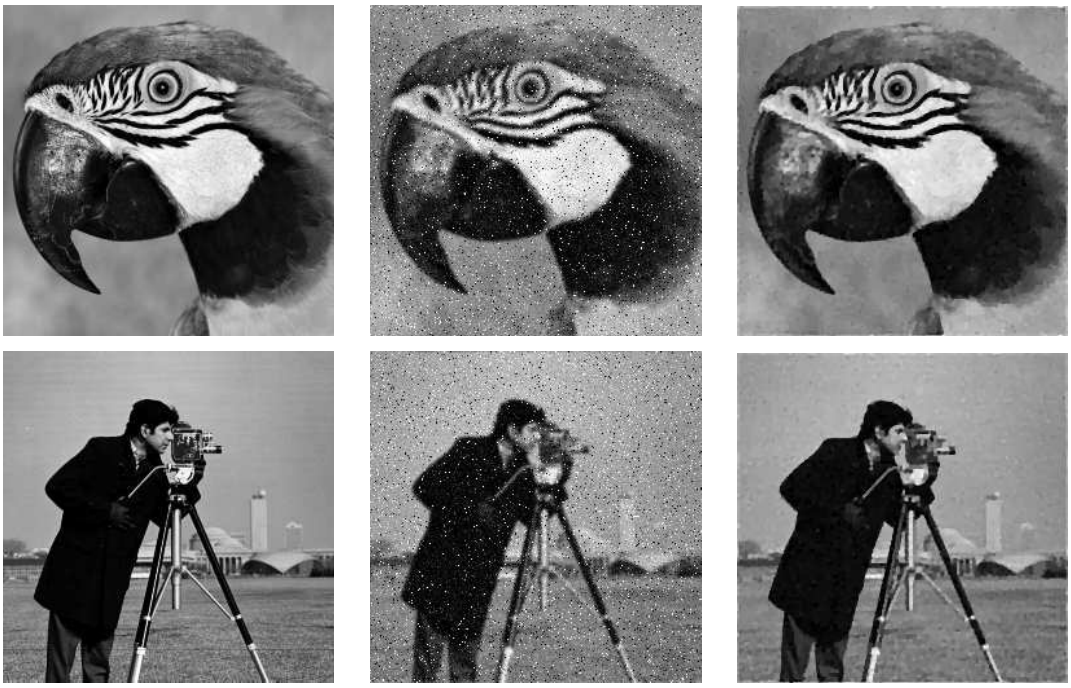


Figure 3.4: Cauchy noise image deblurring datasets: original objects (left), blurred and noisy images (middle) and VMILAn reconstructions (right).

gradient of f_0

$$\nabla f_0(x) = \lambda H^T \frac{(Hx-g)}{\gamma^2 + (Hx-g)^2},$$

we deduce that ∇f_0 is Lipschitz continuous with $L(f_0) \leq \gamma^{-2}\|H\|\|H^T\|$ as upper bound for the smallest Lipschitz constant.

We consider two datasets borrowed by [134, Section 5.2]. In particular the operator H is associated to a Gaussian blur with a window size 9×9 and standard deviation equal to 1, while γ has been set equal to 0.02. We report the true images and the distorted ones in figure 3.4. The regularization parameter λ has been fixed equal to 0.35. We applied VMILAn by computing the proximal point $\tilde{y}^{(k)}$ inexactly by means of the FISTA algorithm as in Section 3.3.1. As in the previous test, we consider three different choices for the scaling matrix, i.e. the Euclidean metric and two further nontrivial metrics. Again, all the matrices considered here are diagonal:

MM $(D_k)_{ii}^{-1} = \max\{\min\{(A_k)_{ii}, \mu\}, \frac{1}{\mu}\}$ where the matrix A_k is borrowed by the MM approach and it is given by formula (36) in [47] where $\varepsilon = 0$ and the function ω is set equal to the function ν in the tenth row of Table 1 in [46].

SG $(D_k)^{-1}_{ii} = \max\left\{\min\left\{\frac{x_i^{(k)}}{V_i(x^{(k)})}, \mu\right\}, \frac{1}{\mu}\right\}$, where $V(x^{(k)}) = \lambda H^T s^{(k)}$ with $s_i^{(k)} = \frac{(Hx^{(k)})_i}{\gamma^2 + (Hx^{(k)} - g)_i^2}$ is again chosen by means of the gradient splitting idea already mentioned in Section 3.3.1. Note that the positivity of $V(x^{(k)})$ is ensured by the non-negative constraints and the properties of the blurring operator.

I $D_k = I_n$.

The other parameters are set exactly as in Section 3.3.1.
In figure 3.5 we show the relative distance between the objective function values and the limit value f^* computed by 5000 iterations of VMILAn with the MM metric. The benefits gained by using a variable metric are quite evident in terms of both number of iterations and computational time.

As further benchmark we include in our comparison also the method VMFB where the majorant function is computed according to Lemma 5.1 in [47] and [46, Table 1].

Finally, to appreciate the validity of VMILAn as restoration method, in table 3.1 we report the values of the peak signal-to-noise ratio (PSNR) related to the approximated solutions compared to the values shown in [134] corresponding to the same two datasets. The PSNR is widely used in the literature to measure the image quality and is defined as

$$\mathrm{PSNR}(x) = 10 \log_{10} \frac{n|\max(x) - \min(x)|^2}{\|x_{\mathrm{true}} - x\|^2},$$

where $x_{\mathrm{true}} \in \mathbb{R}^n$ is the true object.

	Data	VMILAn(I)	VMILAn(SG)	VMILAn(MM)	VMFB	[134]
Parrot	18.23	26.67	26.70	26.71	26.62	26.79
Cameraman	18.29	25.90	26.41	26.52	25.82	26.72

Table 3.1: PSNR values obtained by VMILAn in solving the Cauchy noise image restoration problems.

The PSNR values presented in Table 3.1 allow to say that the performances of VMILAn are comparable to those of the reference approach [134]. The reconstructed images obtained with VMILAn (scal = MM) and related to the PSNR reported in Table 3.1 are shown in the right panel of figure 3.4.

3.3.3 Linear diffusion based image compression

For the next numerical experience, we address the problem of linear diffusion based image compression considered in [105], which consists in finding the optimal interpolation points for the compression procedure (see also [69, 85]). In particular, the problem has been reformulated

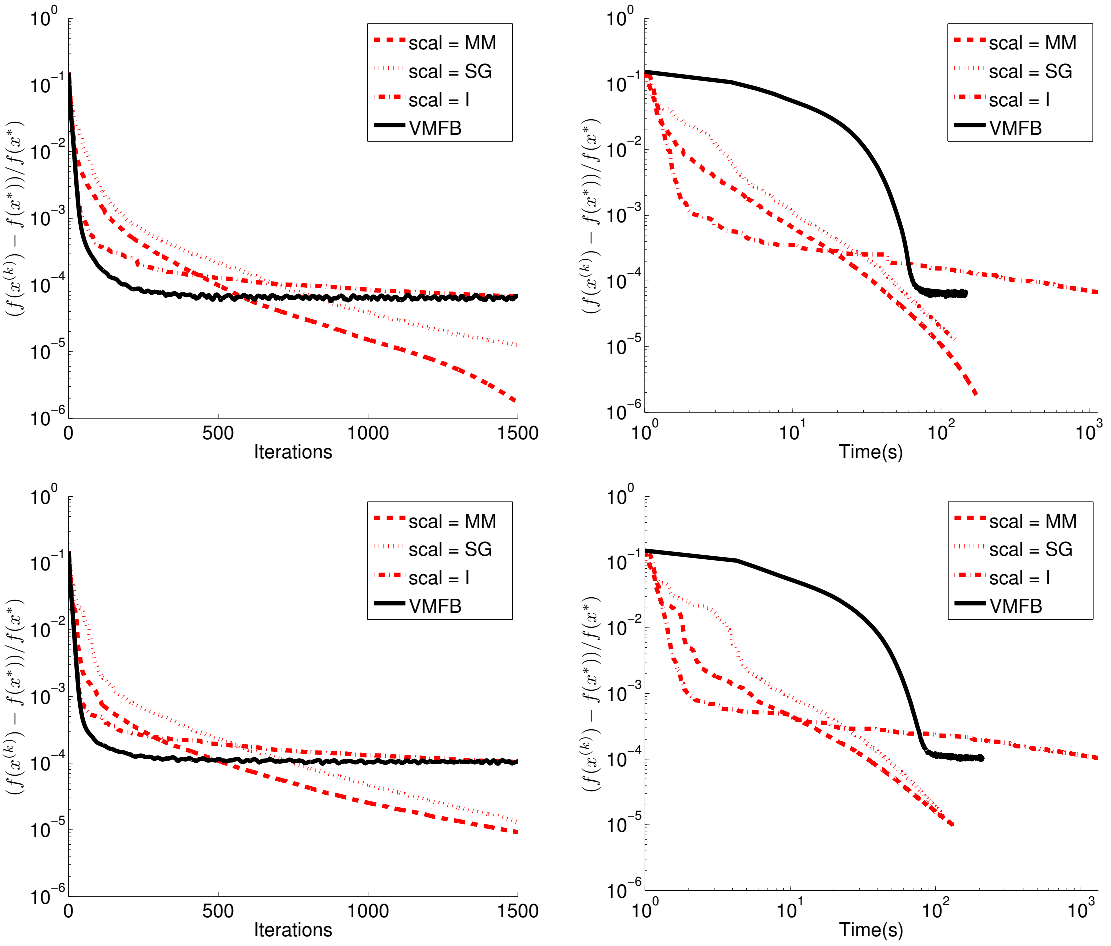


Figure 3.5: Relative decrease of the objective function toward the minimum value with respect to the iteration number (left) and computational time in seconds (right) for the Cauchy noise image restoration datasets: parrot (first row) and cameraman (second row).

in [85, 105] as follows

$$\begin{aligned} &\min_{u,c} \ \frac{1}{2}\|u - u^{(0)}\|_2^2 + \lambda\|c\|_1 \\ &\text{s.t. } C(u - u^{(0)}) - (I_n - C)L_n u = 0, \end{aligned} \tag{3.84}$$

where $u^{(0)} \in \mathbb{R}^n$ denotes the original image, $c \in \mathbb{R}^n$ is the so-called inpainting mask and represents the unknown weights to be assigned to each pixel in the compression step, $C = \text{diag}(c) \in \mathbb{R}^{n\times n}$, $u \in \mathbb{R}^n$ is the image to be reconstructed and $L_n \in \mathbb{R}^{n\times n}$ is the Laplacian operator. Such a problem is nonconvex due to the nonconvexity of the equality constraint, from which the image u can be rewritten as

$$u = A^{-1}Cu^{(0)}$$

with $A = C + (C - I_n)L_n$. If we substitute the above equation into (3.84), we obtain an equivalent optimization problem which depends only on the inpainting mask c. Unlike in [105], we also force the object c to satisfy a certain set of constraints, by adding the indicator function $\iota_{\mathcal{C}}$ to the objective function:

$$\min_{c\in\mathbb{R}^n} \frac{1}{2}\|A^{-1}Cu^{(0)} - u^{(0)}\|_2^2 + \lambda\|c\|_1 + \iota_{\mathcal{C}}(c). \tag{3.85}$$

As concerns the choice of the feasible set $\mathcal{C}$, although the natural choice would be the cartesian product $[0,1]^n$, in our experiments we observed that better results can be obtained by allowing the inpainting mask to assume values greater than 1, and therefore we chose $\mathcal{C} = [0, 1.5]^n$.

The presence of the non-negativity constraint allows to apply VMILAn by including the term $\lambda\|c\|_1$ in the differentiable part f_0 and setting $f_1(c) = \iota_{\mathcal{C}}(c)$. The proximal operator of f_1 reduces to the projection over the set $\mathcal{C}$ and thus it is computed exactly. Moreover, f is a KL function, being the sum of semi-algebraic functions, and ∇f_0 is Lipschitz continuous. Finally, the boundedness of the feasible set $\mathcal{C}$ guarantees the existence of a limit point. All these facts allow to apply Corollary 3.1 and to state the convergence of the VMILAn sequence to a stationary point of f. Furthermore, since the objective function is overall semi-algebraic, the desingularizing function ϕ at the limit point is of the special form $\phi(t) = (ct^\theta)/\theta$ with $\theta \in (0,1]$ (see Section 2.3.1) and thus, on account of Theorem 3.5, the expected convergence rate of both iterates and function values is at least $\mathcal{O}(1/k^p)$, with $p > 1$.

Since the gradient of f_0 does not suggest any natural decomposition, we consider the nonscaled version of VMILAn by setting $D_k = I_n$ for all k. As concerns the steplength parameter α_k, we used the same strategy described in the previous section by replacing (3.81) with

$$\tilde{g}_j^{(k)} = \begin{cases} 0 & \text{if } c_j^{(k)} \in \{0, 1.5\}, \\ \left[\nabla f_0(c^{(k)})\right]_j & \text{if } c_j^{(k)} \in (0, 1.5) \end{cases}$$

and setting $\alpha_{\max} = 10^5$.

We compare VMILAn with the iPiano algorithm, originally devised in [105] and detailed in Algorithm 8 of Section 2.2.3, which is a forward–backward method with extrapolation whose generated sequence converges to a critical point of (3.85) thanks to the KL property of the objective function. Unlike the choice made for VMILAn, here we followed the implementation of the authors and left the term $\lambda\|c\|_1$ in the f_1 part of the objective function (we tried also the other splitting but we always obtained worse results). All the other parameters defining iPiano have been chosen as suggested in [105]. The test problems are the same used in [105, §5.2.2] and named “trui”, “peppers” and “walter” (see Figure 3.6). In Table 3.2 we report the iteration numbers performed by the two methods together with the corresponding values of the

Test image	Algorithm	Iterations	Obj. func.	Density	MSE
trui	iPiano	1000	21.58	4.97%	17.27
	VMILAn	599	21.50	4.80%	17.95
peppers	iPiano	1000	23.10	5.95%	19.64
	VMILAn	655	23.01	5.81%	19.99
walter	iPiano	1000	10.32	5.10%	8.27
	VMILAn	699	10.23	4.66%	8.55

Table 3.2: Summary of two algorithms for three test images.

objective function, density and mean squared error (MSE) computed by

$$\mathrm{MSE}(u, u^{(0)}) = \frac{1}{n}\sum_{i=1}^{n}(u_i - u_i^0)^2,$$

where $u = A^{-1}Cu^{(0)}$ is the reconstructed image. Moreover, since in this case it seems that the two algorithms do not converge to the same minima, in Figure 3.7 we do not plot the relative distance between the objective function and the minimum but we show the decrease of the objective function with respect to the iteration number and the computational time in seconds.

The behaviour of the steplength α_k and the linesearch parameter λ_k is also shown in Figure 3.8. Concerning the former parameter, it can be seen that the value α_k varies of several order of magnitudes; this is typical of any steplength selection rule which aims at approximating the spectrum of the Hessian of the function, as it is the case of the rule we adopted for VMILAn. Indeed, it looks like the red plots in Figure 3.8 are oscillating between two extreme values, which might be considered as approximations of the reciprocals of some eigenvalues of the Hessian matrix $\nabla^2 f_0(x^{(k)})$.

Finally, in the right column of figure 3.6 the reconstructions obtained with VMILAn are given. As remarked in the previous numerical tests, also in this application VMILAn seems to be competitive if compared to other forward–backward approaches, since it is able to provide comparable reconstructions by performing a lower number of iterations and allowing a reduction of the computational time. In all the experiments described in this section, the first option in (3.25) never occurred.

3.3.4 Student-t regularized image denoising

In this final section, we consider again an application in imaging used in [105] and consisting in an image denoising problem addressed by means of the Markov random field (MRF) model

$$\min_{x\in\mathbb{R}^n}\sum_{i=1}^{N_f}\theta_i\left(\sum_{p=1}^{n}\log(1+(k_i\otimes x)_p^2)\right)+\frac{\rho}{2}\|x-g\|_2^2+\iota_{\mathcal{X}}(x).$$

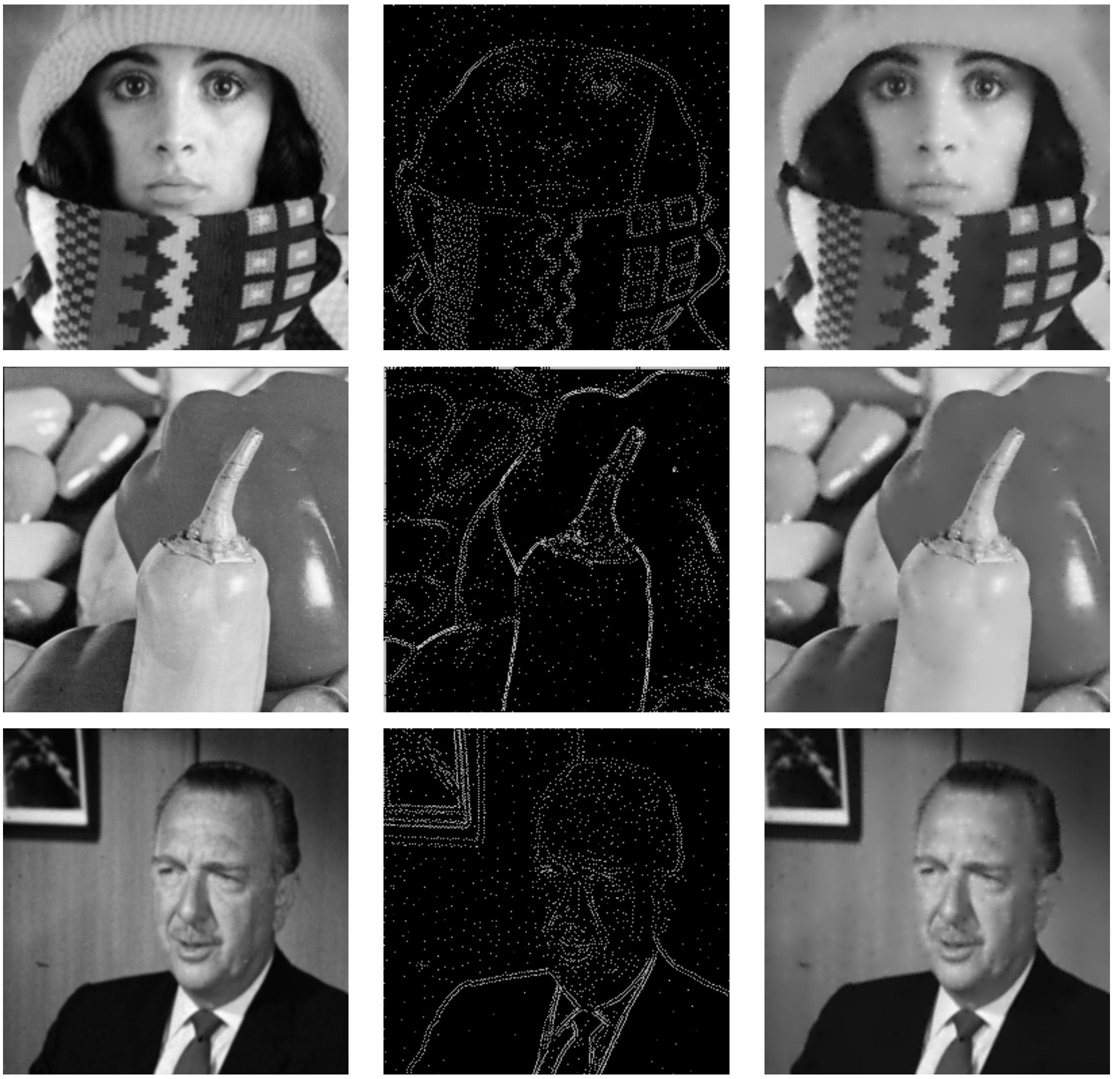


Figure 3.6: Trui (top row), peppers (central row) and walter (bottom row) datasets. Original image (left), inpainting mask (middle) and VMILAn reconstruction (right).

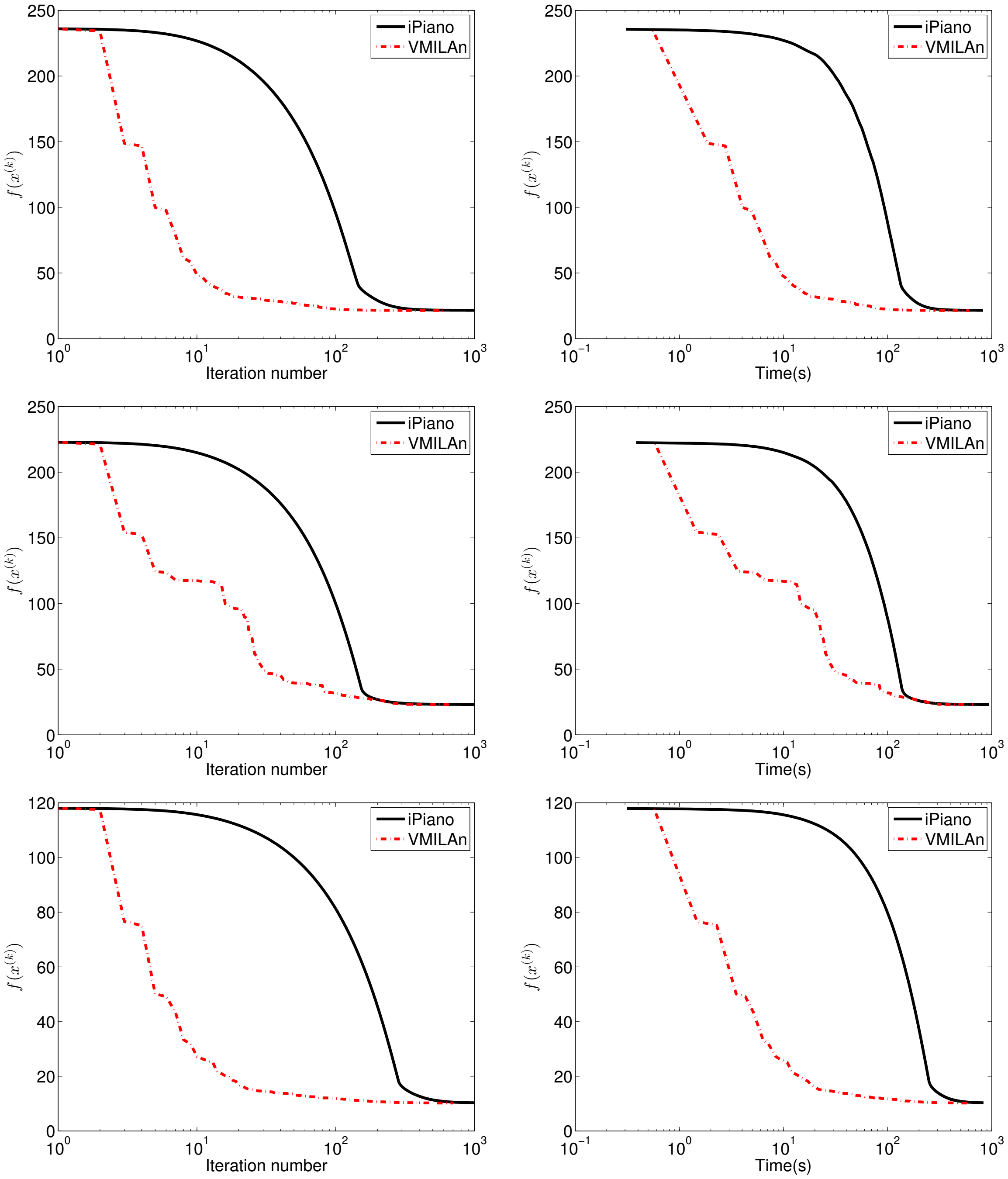


Figure 3.7: Linear diffusion based image compression for the trui (top row), peppers (central row) and walter (bottom row) datasets. Decrease of the objective function with respect to the iteration number (left) and computational time in seconds (right).

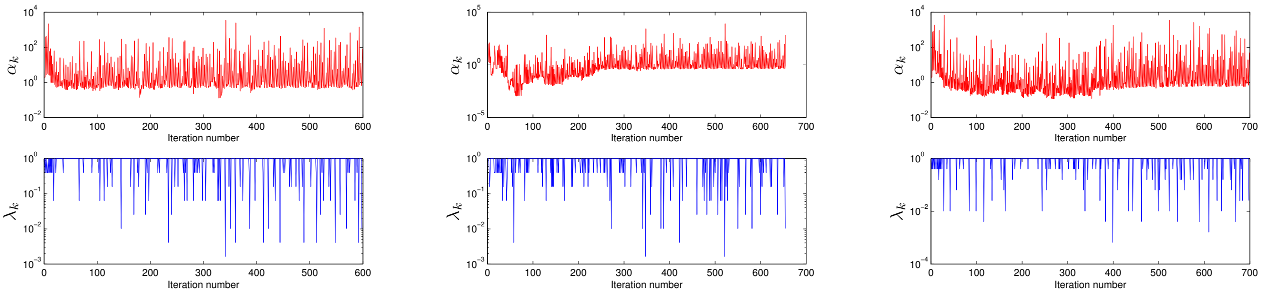


Figure 3.8: Behaviour of the parameters α_k and λ_k with respect to the iteration number for the trui (left column), peppers (central column) and walter (right column) datasets.

Here x and g denote the target and the (Gaussian) noisy images, respectively, θ_i are positive weights, $\otimes$ denotes the two-dimensional convolution and k_i are 7×7 filter kernels learned in [45] by using a bilevel learning approach (weights and filters can be downloaded from [44], together with the instructions to produce the noisy image). As concerns the feasible set $\mathcal{X}$, we force the reconstructed image to belong to the non-negative orthant $\mathbb{R}^n_{\geq 0}$.
The image used for this experiment is the so–called watercastle, and it is shown in Figure 3.8, together with the one obtained by the true image by adding Gaussian noise with standard deviation $\sigma = 25$. The regularization parameter ρ has been fixed equal to 1 as suggested in [105]. We applied VMILAn by choosing f_1 equal to $\iota_{\mathcal{X}}$ and f_0 equal to the remaining part. As reference method we used as in the previous section the iPiano algorithm, in the same settings described by the authors in [105]. Since both algorithms converge to the same solution, as done in Section 3.3.1 we computed the relative distance between the objective function during the iterations and its minimum value, obtained by performing 1000 iterations with iPiano. The corresponding plots with respect to the iteration number and the computational time are shown in Figure 3.10, while the denoised image provided by VMILAn after 250 iterations is given in figure 3.9.
This last experiment confirms the conclusions previously drawn for VMILAn in the image compression application, since also in this case our method behaves similarly to iPiano in terms of both number of iterations required to minimize the objective function and average cost per iteration.

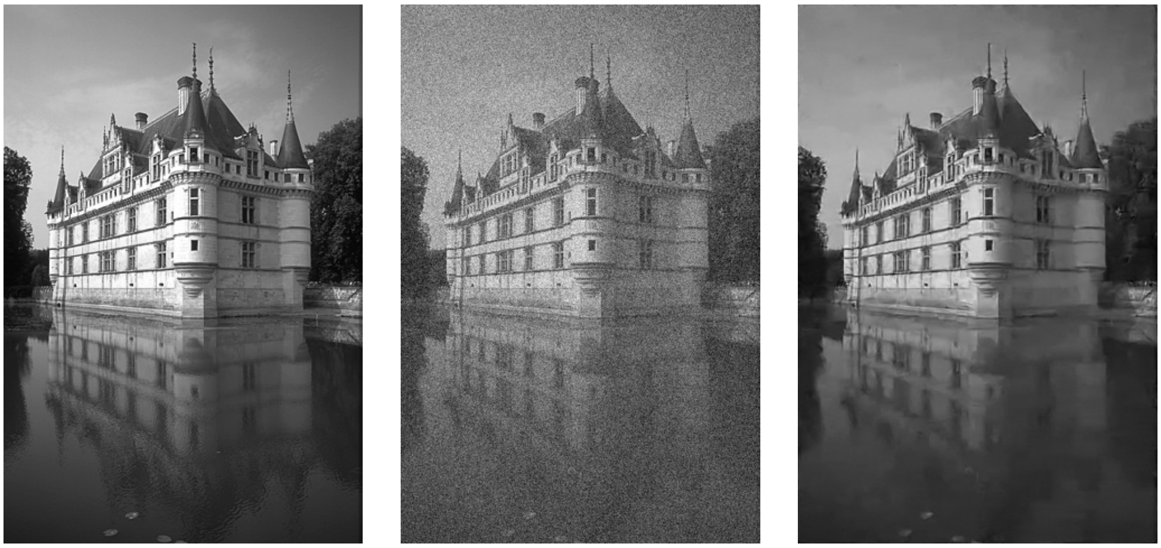


Figure 3.9: Watercastle test problem: original object (left), blurred and noisy image (middle), and VMILAn reconstruction (right).

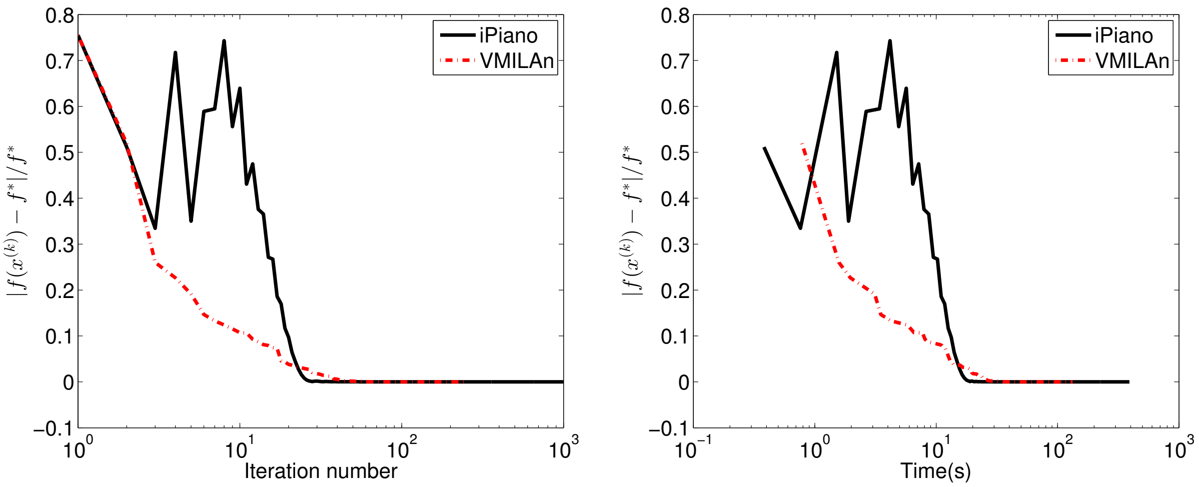


Figure 3.10: Student-t regularized image denoising for the watercastle dataset. Decrease of the objective function with respect to the iteration number (left) and computational time in seconds (right).

Chapter 4

Phase estimation in differential interference contrast (DIC) microscopy

In this chapter we are interested in a specific application in the field of optical microscopy, which can be suitably addressed by the line–search based methods devised in the previous chapter. In particular, we consider the problem of phase estimation from color images acquired in difference interference contrast (DIC) microscopy. In the last forty years, DIC microscopy has gained popularity in biomedical research as an effective optical microscopy technique used to observe unstained transparent specimens under a transmitted-light configuration. DIC microscopes are able to provide contrast to images by exploiting the phase shifts in light induced by the transparent specimens (also called phase objects) while passing through them. One disadvantage of DIC microscopy is that the observed images cannot be easily used for topographical and morphological interpretation, because the changes in phase of the light are hidden in the intensity image. It is then of vital importance to recover the specimen's phase function from the observed DIC images. The problem can be reformulated in mathematical terms as an optimization problem which, unfortunately, is highly nonconvex and presents multiple local minima. For that reason, there is the need of efficient computational methods aimed at recovering quantitative information on the DIC phase function. So far, only few works have dealt with this problem in the literature [117, 118, 115, 116].

Our aim is to exploit the line–search based methods developed in the previous chapter in the context of DIC imaging. In particular, we address the DIC phase estimation problem with a gradient method, equipped with an Armijo line–search and a non standard selection rule for the steplength, and a non-scaled version of Algorithm VMILAn presented in Chapter 3. Furthermore, we revisit the state-of-the-art optimization method for phase estimation in DIC microscopy providing implementation details, showing possible pitfalls and comparing its performances with standard conjugate gradient algorithms. Finally, we show that, in numerical simulations with simulated datasets, the two proposed optimization strategies are able to provide accurate reconstructions of the phase in a lower computational time.

The chapter is organized as follows. In Section 4.1, we provide the details concerning the DIC model and we consider the minimization problem of the functional given by the sum of the maximum likelihood term and a (possibly smoothed version of) total variation regularizer, studying its analytical properties and proving the existence of minimum points. In Section 4.2, we present the proposed methods and revisit the state-of-the-art optimization method for phase estimation in DIC microscopy. In Section 4.3, numerical experience on simulated datasets is presented.

4.1 Model and problem formulation

The technique of interest in this chapter is Differential Interference Contrast (DIC) microscopy, designed by Allen, David and Nomarski [4] to overcome the inability to image unstained transparent biological specimens, which is typical of bright-field microscopes, while avoiding at the same time the halo artifacts of other techniques designed for the same purpose, such as phase contrast.

4.1.1 The DIC model

DIC microscopy works under the principle of dual-beam interference of polarized light, as depicted in Figure 4.1. Coherent light coming from a source is passed through a polarizer lens. Every incident ray of polarized light is splitted by a Nomarski prism placed at the front focal plane of the condenser. This splitting produces two wave components – ordinary and extraordinary – such that their corresponding electromagnetic fields are orthogonal and separated at a fixed shear distance $2\Delta x$ along a specific shear direction, whose angle τ_k formed with the x-axis is denominated shear angle. The specimen is sampled by the pair of waves; if they pass through a region where there is a gradient in the refractive index, the waves will be differentially shifted in phase. After this, they will reach a second Normarski prism placed at the back focal plane of the objective lens. This prism introduces an additional phase shift, called the bias retardation and indicated with $2\Delta\theta$, which helps to improve the contrast of the observed image and to give the shadow-cast effect characteristic of DIC images (see Figure 4.2). The interference of the two sheared and phase shifted waves occurs inside this prism and, thus, the two waves are recombined into a single beam that goes through a second polarizer lens called the analyzer. Further details on the DIC working principle can be found in the work of Murphy [100] and Mehta et al [98].

The observed images will have a uniform gray background on regions where there are no changes in the optical path, whereas they will have dark shadows and bright highlights where there are phase gradients in the direction of shear, having a 3-D relief-like appearance (see

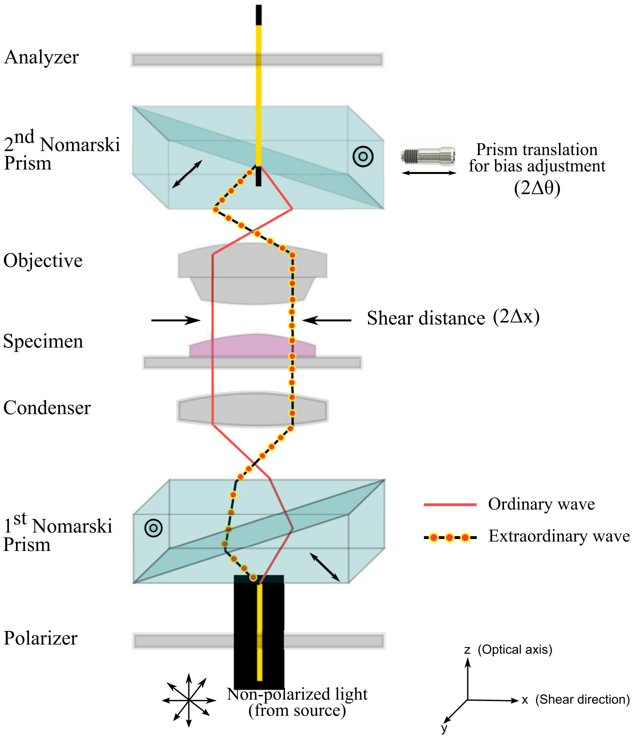


Figure 4.1: Transmitted-light Nomarski DIC microscope. The difference of colors of the ordinary and extraordinary waves indicates that their electromagnetic fields are orthogonal to each other.

Figure 4.2). It is important to note that the shadows and highlights indicate the signs and slope of phase gradients in the specimen, and not necessarily indicate high or low spots [4].

In this paper we consider the polychromatic rotational-diversity model [13], which is an extension of the model presented in [115] to color image acquisition. This model assumes that K color images are acquired by rotating the specimen K times with respect to the shear axis, which results in K rotations of the amplitude point spread function. Typically K equals 2 and the difference between the two angles is $\pi/2$. Actually, for a given shear angle τ_k, the acquired image k is related to the directional derivative of the object along the direction τ_k [118]. Then the 2D image can be reconstructed from two orthogonal directional derivatives [115]. In this configuration, the relation between the acquired images and the unknown true phase ϕ is given by

$$(o_{k,\lambda_\ell})_j = a_1 \left|(h_{k,\lambda_\ell} \otimes e^{-i\phi/\lambda_\ell})_j\right|^2 + (\eta_{k,\lambda_\ell})_j, \tag{4.1}$$

for $k = 1, \ldots, K$, $\ell = 1, 2, 3$, $j \in \chi$, where

- k is the index of the angles τ_k that the shear direction makes with the horizontal axis

[115], ℓ is the index denoting one of the three RGB channels and $j = (j_1, j_2)$ is a 2D–index varying in the set $\chi = \{1, \dots, M\} \times \{1, \dots, P\}$, M and P meaning the size of the acquired image, which is determined by the resolution of the CCD detector of the microscope, with typical value of 1388 × 1040 pixels;

- λ_ℓ is the $\ell-$th illumination wavelength, which is assumed to be rational. The object is illuminated with white light, whose wavelengths range from 400 nm to 700 nm. The digital acquisition system of the microscope comprises a color bandpass filter which isolates the RGB wavelengths, acquired separately by the CCD detector [100]. Since it is selected a narrow band for each color, we use the mean wavelength at each band. Without loss of generality, this mean value can be considered as a rational number. In particular, in our experiments we will set $\lambda_1 = 0.65$, $\lambda_2 = 0.55$ and $\lambda_3 = 0.45$ as values for the red, green and blue wavelengths, respectively;

- $o_{k,\lambda_\ell} \in \mathbb{R}^{MP}$ is the $\ell-$th color component of the $k-$th discrete observed image $o_k = (o_{k,\lambda_1}, o_{k,\lambda_2}, o_{k,\lambda_3}) \in \mathbb{R}^{MP\times 3}$;

- $\phi \in \mathbb{R}^{MP}$ is the unknown phase vector and $e^{-i\phi/\lambda_\ell} \in \mathbb{C}^{MP}$ stands for the vector defined by $(e^{-i\phi/\lambda_\ell})_j = e^{-i\phi_j/\lambda_\ell}$;

- $h_{k,\lambda_\ell} \in \mathbb{C}^{MP}$ is the discretization of the continuous DIC point spread function [118, 74] corresponding to the illumination wavelength λ_ℓ and rotated by the angle τ_k , i.e.,

$$h_{k,\lambda_\ell}(x, y) = \frac{1}{2}\left[e^{-i\Delta\theta} p_{\lambda_\ell}\left(R_k \cdot (x - \Delta x, y)^T\right) - e^{i\Delta\theta} p_{\lambda_\ell}\left(R_k \cdot (x + \Delta x, y)^T\right)\right], \qquad (4.2)$$

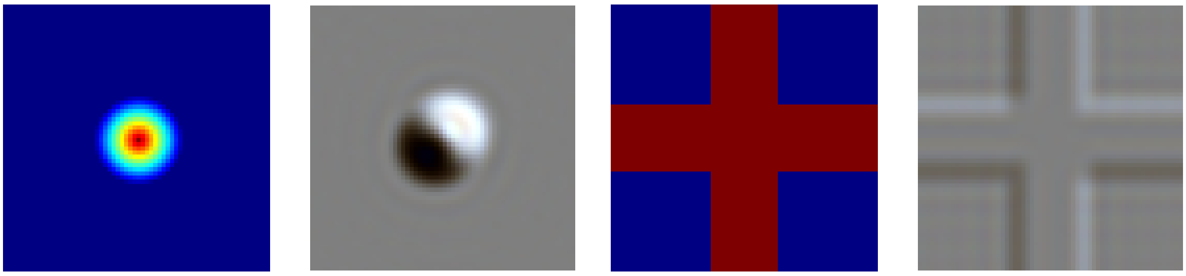
  where $p_{\lambda_\ell}(x, y)$ is the coherent PSF of the microscope's objective lens for the wavelength λ_ℓ, which is given by the inverse Fourier transform of the disk support function of amplitude 1 and radius equal to the cutoff frequency $f_c = NA/\lambda_\ell$ [118], being NA the numerical aperture of the objective lens, and R_k is the rotation matrix which rotates the coordinates according to the shear angle τ_k;

- $h_1 \otimes h_2$ denotes the 2D convolution between the two $M \times P$ images h_1, h_2, extended with periodic boundary conditions;

- $\eta_{k,\lambda_\ell} \in \mathbb{R}^{MP}$ is the noise corrupting the data, which is assumed to be a realization of a Gaussian random vector with mean $\mathbf{0} \in \mathbb{R}^{MP}$ and covariance matrix $\sigma^2 I_{(MP)^2}$, where $I_{(MP)^2}$ is the identity matrix of size $(MP)^2$;

- $a_1 \in \mathbb{R}$ is a constant which corresponds to closing the condenser aperture down to a single point.

The problem with the acquired DIC images is their high sensitivity to the shear direction and the chosen value of the bias, as we see in Figure 4.1.2 and 4.1.2. Thus, topological and morphological information might be hidden or difficult to interpret in the final acquired image.



(a) (b) (c) (d)

Figure 4.2: Phase functions of two phantom specimens and corresponding noiseless DIC color images: (a) phase function of the "cone" object, (b) DIC image of the cone, (c) phase function of the "cross" object, (d) DIC image of the cross. The images have been computed by using model (4.1) and setting the shear to $2\Delta x = 0.6$ μm, the bias to $2\Delta\theta = \pi/2$ rad and the shear angle to $\tau = \pi/4$ rad.

On account of that, one is more interested in the approximation of the phase ϕ, which is independent of the acquisition direction and the bias value.

4.1.2 Optimization problem

The phase reconstruction problem consists in finding an approximation of the unknown phase vector ϕ from the observed RGB images $o_1, \ldots, o_K$. Let us first address this problem by means of the maximum likelihood (ML) approach (see Appendix A). Since the $3K$ images o_{k,λ_ℓ} are corrupted by Gaussian noise, then the negative log likelihood of each image is a least-squares measure, which is nonlinear due to the presence of the exponential in (4.1). If we assume white Gaussian noise, statistically independent of the data, the negative log likelihood of the problem is the sum of the negative log likelihoods of the different images, namely the following fit-to-data term

$$J_0(\phi) = \sum_{\ell=1}^{3}\sum_{k=1}^{K}\sum_{j\in\chi}\left[(o_{k,\lambda_\ell})_j - a_1\left|(h_{k,\lambda_\ell}\otimes e^{-i\phi/\lambda_\ell})_j\right|^2\right]^2. \tag{4.3}$$

Then the ML approach to the phase reconstruction inverse problem consists in the minimization of the function in (4.3):

$$\min_{\phi\in\mathbb{R}^{MP}} J_0(\phi). \tag{4.4}$$

In the next result, we collect some properties of J_0 that will be useful hereafter.

Lemma 4.1. *Let $J_0 : \mathbb{R}^{MP} \to \mathbb{R}$ be defined as in* (4.3)*. Then the following facts hold.*

(i) There exists $T > 0$ such that J_0 is periodic of period T with respect to each variable, i.e. for any $j \in \chi$, defining $\boldsymbol{e}_j = (\delta_{j,r})_{r\in\chi} = (0, \ldots, 0, 1, 0, \ldots, 0) \in \mathbb{R}^{MP}$, where $\delta_{j,r}$ is the

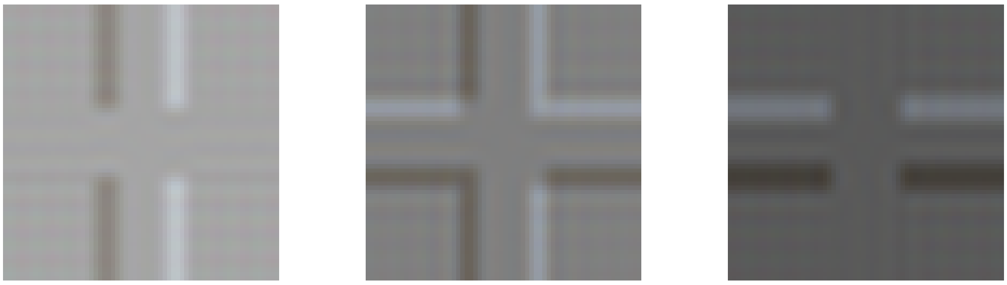


Figure 4.3: DIC images are direction-sensitive. From left to right: cross object computed as in Figure 4.2 acquired at shear angles $\tau = 0, \pi/4, \pi/2$ rad respectively. According to the direction, the information on one of the two crossing bars might be lost.

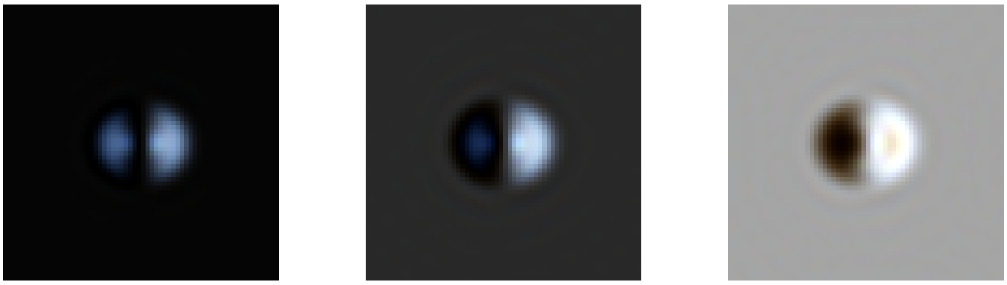


Figure 4.4: DIC images are bias-sensitive. From left to right: cone object computed as in Figure 4.2 acquired at bias value $2\Delta\theta = 0, \pi/6, \pi/2$ rad respectively. According to the bias value, the shape of the cone might be unrecognizable.

Kronecker delta, it holds

$$J_0(\phi + T\boldsymbol{e}_j) = J_0(\phi), \quad \forall\ \phi \in \mathbb{R}^{MP}. \tag{4.5}$$

(ii) $J_0(\phi + c\boldsymbol{e}) = J_0(\phi)$, $\forall\ c \in \mathbb{R}$, *where* $\boldsymbol{e} \in \mathbb{R}^{MP}$ *is the vector of all ones.*

(iii) J_0 *is an analytic function on* $\mathbb{R}^{MP}$ *and therefore* $J_0 \in C^\infty(\mathbb{R}^{MP})$*.*

Proof. (i) Fix $j \in \chi$, $\ell \in \{1, 2, 3\}$ and consider the exponential in (4.3). Then for all $r \in \chi$

$$\left(e^{-i(\phi + 2\pi\lambda_\ell \boldsymbol{e}_j)/\lambda_\ell}\right)_r = \begin{cases} e^{-i\phi_r/\lambda_\ell} & , \ r \neq j \\ e^{-i[(\phi_j/\lambda_\ell)+2\pi]} = e^{-i\phi_r/\lambda_\ell} & , \ r = j \end{cases} = (e^{-i\phi/\lambda_\ell})_r, \tag{4.6}$$

where the equality inside the curly bracket is due to the periodicity of the complex exponential. Then, for a fixed $\ell \in \{1, 2, 3\}$, the expression given in (4.3) without the sum in ℓ is $2\pi\lambda_\ell$ periodic w.r.t. the variable ϕ_j. This means that J_0 is the sum of three periodic functions of variable ϕ_j

whose periods are $2\pi\lambda_1$, $2\pi\lambda_2$ and $2\pi\lambda_3$ respectively. By recalling that the sum of two periodic functions is periodic if the ratio of the periods is a rational number, we can conclude that J_0 is periodic, as we have $\frac{\lambda_\ell}{\lambda_{\ell'}}$ rational for all $\ell, \ell' \in \{1,2,3\}$.

(ii) Set $J_{\ell,k,j}(\phi) = \left|(h_{k,\lambda_\ell} \otimes e^{-i\phi/\lambda_\ell})_j\right|^2 = \left|\sum_{r\in\chi}(h_{k,\lambda_\ell})_r e^{-i(\phi_{j-r})/\lambda_\ell}\right|^2$. If the thesis holds for $J_{\ell,k,j}$, then it holds also for J_0. We have

$$\begin{aligned} J_{\ell,k,j}(\phi + c\boldsymbol{e}) &= \left|\sum_{r\in\chi}(h_{k,\lambda_\ell})_r e^{-i(\phi_{j-r}+c)/\lambda_\ell}\right|^2 = \left|e^{-ic/\lambda_\ell}\sum_{r\in\chi}(h_{k,\lambda_\ell})_r e^{-i(\phi_{j-r})/\lambda_\ell}\right|^2 \\ &= \left|e^{-ic/\lambda_\ell}\right|^2 \left|\sum_{r\in\chi}(h_{k,\lambda_\ell})_r e^{-i(\phi_{j-r})/\lambda_\ell}\right|^2 = J_{\ell,k,j}(\phi). \end{aligned} \tag{4.7}$$

(iii) If $J_{\ell,k,j}$ is an analytic function on $\mathbb{R}^{MP}$, then J_0 is given by sums and compositions of analytic functions and thus it is itself analytic [88, Propositions 1.6.2 and 1.6.7]. Hence we focus on $J_{\ell,k,j}$. Since $(h_{k,\lambda_\ell})_r \in \mathbb{C}$, it can be expressed in its trigonometric form $(h_{k,\lambda_\ell})_r = \rho_r e^{i\theta_r}$, with $\rho_r \in \mathbb{R}_{\geq 0}$, $\theta_r \in [0, 2\pi)$. Then we can rewrite $J_{\ell,k,j}$ as follows

$$\begin{aligned} J_{\ell,k,j}(\phi) &= \left|\sum_{r\in\chi}\rho_r e^{i[\theta_r - (\phi_{j-r}/\lambda_\ell)]}\right|^2 = \\ &= \left|\sum_{r\in\chi}\rho_r \cos(\theta_r - (\phi_{j-r}/\lambda_\ell)) + i\sum_{r\in\chi}\rho_r \sin(\theta_r - (\phi_{j-r}/\lambda_\ell))\right|^2 = \\ &= \left(\sum_{r\in\chi}\rho_r \cos(\theta_r - (\phi_{j-r}/\lambda_\ell))\right)^2 + \left(\sum_{r\in\chi}\rho_r \sin(\theta_r - (\phi_{j-r}/\lambda_\ell))\right)^2. \end{aligned}$$

We now observe that the function $J_{\ell,k,j}$ contains $\sin(\theta_r - (\phi_{j-r}/\lambda_\ell))$ and $\cos(\theta_r - (\phi_{j-r}/\lambda_\ell))$, which are both analytic functions with respect to the single variable ϕ_{j-r} and thus also with respect to ϕ, and the square function $(\cdot)^2$, which is also analytic. Since $J_{\ell,k,j}$ is given by sums and compositions of these functions, it is analytic. □

Problem (4.4) admits infinitely many solutions, as stated in the following theorem.

Theorem 4.1. *J_0 admits at least one global minimum point. Furthermore, if $\psi \in \mathbb{R}^{MP}$ is a global minimizer of J_0, then also $\{\psi + c\boldsymbol{e} : c \in \mathbb{R}\} \cup \{\psi + mT\boldsymbol{e}_j : j \in \chi, m \in \mathbb{Z}\}$ are global minimizers of J_0.*

Proof. Let $\Omega = [0, T]^{MP} \subset \mathbb{R}^{MP}$. Point (iii) of Lemma 4.1 ensures that J_0 is continuous on Ω, thus from the extreme value theorem J_0 admits at least one minimum point ψ on Ω. By contradiction, assume that there exists $\phi \in \mathbb{R}^{MP} \setminus \Omega$ such that $J_0(\phi) < J_0(\psi)$. Let $I \subset \chi$ be the subset of indices such that $\{\phi_s\}_{s\in I}$ is the set of all components of ϕ which belong to

$\mathbb{R} \setminus [0,T]$ and $\{m_s\}_{s\in I} \subset \mathbb{Z} \setminus \{1\}$ is the set of integers such that $\phi_s \in [(m_s-1)T, m_sT]$. Define $\bar{\phi} = \phi - \sum_{s\in I}(m_s-1)T\boldsymbol{e}_s \in \Omega$. By periodicity of J_0 w.r.t. the variables ϕ_s, $s \in I$, we obtain

$$J_0(\bar{\phi}) = J_0(\phi) < J_0(\psi). \tag{4.8}$$

Therefore, we have found a point $\bar{\phi} \in \Omega$ such that $J_0(\bar{\phi}) < J_0(\psi)$, where ψ is a minimum point on Ω. This is absurd, hence ψ is a global minimizer for J_0. The second part of the thesis follows from points (i)-(ii) of Lemma 4.1. □

Theorem 4.1 asserts that the solution to problem (4.4) is not unique and it may be determined only up to an unknown real constant or to multiples of the period T w.r.t. any variable ϕ_j. Furthermore, since J_0 is periodic, it is a nonconvex function of the phase ϕ, thus it may admit several local minima as well as saddle points. In the light of these considerations, we can conclude that (4.4) is a severely ill-posed problem, which requires regularization in order to impose some a priori knowledge on the unknown phase. In particular, we propose to solve the following regularized optimization problem

$$\min_{\phi\in\mathbb{R}^{MP}} J(\phi) \equiv J_0(\phi) + J_{TV}(\phi), \tag{4.9}$$

where J_0 is the least-squares distance defined in (4.3) and J_{TV} is the smooth total variation functional (also known as hypersurface potential - HS) defined as [2, 19]

$$J_{TV}(\phi) = \mu \sum_{j\in\chi} \sqrt{((\mathcal{D}\phi)_j)_1^2 + ((\mathcal{D}\phi)_j)_2^2 + \delta^2}, \tag{4.10}$$

where $\mu > 0$ is a regularization parameter, the discrete gradient operator $\mathcal{D} : \mathbb{R}^{MP} \longrightarrow \mathbb{R}^{2MP}$ is set through the standard finite difference with periodic boundary conditions

$$(\mathcal{D}\phi)_{j_1,j_2} = \begin{pmatrix} ((\mathcal{D}\phi)_{j_1,j_2})_1 \\ ((\mathcal{D}\phi)_{j_1,j_2})_2 \end{pmatrix} = \begin{pmatrix} \phi_{j_1+1,j_2} - \phi_{j_1,j_2} \\ \phi_{j_1,j_2+1} - \phi_{j_1,j_2} \end{pmatrix}, \ \phi_{M+1,j_2} = \phi_{1,j_2}, \ \phi_{j_1,P+1} = \phi_{j_1,1}$$

and the additional parameter $\delta \geq 0$ plays the role of a threshold for the gradient of the phase. Obviously J_{TV} reduces to the standard TV functional [132] by setting $\delta = 0$. The choice of this kind of regularization term instead of the first-order Tikhonov one used e.g. in [115, 116] lies in the capability of the HS regularizer to behave both as a Tikhonov-like regularization in regions where the gradient assumes small values (w.r.t. δ), and as an edge-preserving regularizer in regions where the gradient is very large, as it happens in the neighborhood of jumps in the values of the phase.

Problem (4.9) is still a difficult nonconvex optimization problem and, when $\delta = 0$, it is also nondifferentiable. Some properties of the objective function J are now reported.

Lemma 4.2. *Let $J : \mathbb{R}^{MP} \to \mathbb{R}$ be defined as in* (4.9)*. Then:*

(i) $J(\phi + c\boldsymbol{e}) = J(\phi)$, $\forall\ c \in \mathbb{R}$.

(ii) If $\delta > 0$, *then* $J \in C^\infty(\mathbb{R}^{MP})$ *and* ∇J *is Lipschitz continuous, namely there exists* $L > 0$ *such that*

$$\|\nabla J(\phi) - \nabla J(\psi)\|_2 \le L\|\phi - \psi\|_2, \quad \forall \phi, \psi \in \mathbb{R}^{MP}. \tag{4.11}$$

Proof. (i) We have already proved in point (ii) of Lemma 4.1 that the property holds for J_0. Since it is immediate to check that $(\mathcal{D}(\phi + c\boldsymbol{e}))_{j_1,j_2} = (\mathcal{D}\phi)_{j_1,j_2}$, the property is true also for J_{TV} and thus for J.
(ii) Point (iii) of Lemma 4.1 states that $J_0 \in C^\infty(\mathbb{R}^{MP})$ and the same property holds for J_{TV}, hence J is the sum of two $C^\infty(\mathbb{R}^{MP})$ functions.
It is known that ∇J_{TV} is $L_{TV}-$Lipschitz continuous with $L_{TV} = 8\mu/\delta^2$ [52]. We prove that also ∇J_0 is Lipschitz continuous. If we introduce the residual image $r_{k,\lambda_\ell} = \left|(h_{k,\lambda_\ell} \otimes e^{-i\phi/\lambda_\ell})\right|^2 - o_{k,\lambda_\ell}$ and fix $s \in \chi$, the partial derivative of J_0 with respect to ϕ_s is given by

$$\frac{\partial J_0(\phi)}{\partial \phi_s} = \sum_{\ell=1}^{3}\sum_{k=1}^{K}\sum_{j\in\chi}\frac{4}{\lambda_\ell}(r_{k,\lambda_\ell})_j \operatorname{Im}\left\{e^{-i\phi_s/\lambda_\ell}(h_{k,\lambda_\ell})_{j-s}\overline{(h_{k,\lambda_\ell} \otimes e^{-i\phi/\lambda_\ell})_j}\right\}, \tag{4.12}$$

where $\operatorname{Im}(\cdot)$ denotes the imaginary part of a complex number. As concerns the entries of the Hessian $\nabla^2 J_0$, the second derivative w.r.t. ϕ_s, ϕ_t $(s, t \in \chi)$ is given by

$$\begin{aligned}\frac{\partial^2 J_0(\phi)}{\partial \phi_t \partial \phi_s} = 4\sum_{\ell=1}^{3}\sum_{k=1}^{K}\sum_{j\in\chi} & \frac{2}{\lambda_\ell^2}\operatorname{Im}\{\vartheta_s\}\operatorname{Im}\{\vartheta_t\} + \\ & \frac{(r_{k,\lambda_\ell})_j}{\lambda_\ell^2}\operatorname{Re}\left\{e^{i(\phi_t-\phi_s)/\lambda_\ell}(h_{k,\lambda_\ell})_{j-s}\overline{(h_{k,\lambda_\ell})_{j-t}} - \delta_{s,t}\vartheta_s\right\},\end{aligned} \tag{4.13}$$

where $\vartheta_p = e^{-i\phi_p/\lambda_\ell}(h_{k,\lambda_\ell})_{j-p}\overline{(h_{k,\lambda_\ell} \otimes e^{-i\phi/\lambda_\ell})_j}$ $(p \in \chi)$, $\operatorname{Re}(\cdot)$ denotes the real part of a complex number and $\delta_{s,t}$ is the Kronecker delta. By using the triangle inequality and the fact that $|e^{-i\phi_r/\lambda_\ell}| = 1$, the following inequality hold:

$$|\vartheta_p| \le |(h_{k,\lambda_\ell})_{j-p}|\sum_{r\in\chi}|(h_{k,\lambda_\ell})_r|. \tag{4.14}$$

By applying the triangle inequality, the fact that $|e^{-i\phi_r/\lambda_\ell}| = 1$, $|\operatorname{Im}(z)| \le |z|$ and $|\operatorname{Re}(z)| \le |z|$ for any $z \in \mathbb{C}$ and inequality (4.14) to (4.13), we obtain the following bound on the second derivative of J_0:

$$\begin{aligned}\left|\frac{\partial^2 J_0(\phi)}{\partial \phi_t \partial \phi_s}\right| \le 4\sum_{\ell=1}^{3}\sum_{k=1}^{K}\sum_{j\in\chi} & \frac{2}{\lambda_\ell^2}|(h_{k,\lambda_\ell})_{j-s}||(h_{k,\lambda_\ell})_{j-t}|\left(\sum_{r\in\chi}|(h_{k,\lambda_\ell})_r|\right)^2 + \\ & \frac{|(r_{k,\lambda_\ell})_j|}{\lambda_\ell^2}\left\{|(h_{k,\lambda_\ell})_{j-s}||(h_{k,\lambda_\ell})_{j-t}| + |(h_{k,\lambda_\ell})_{j-s}|\sum_{r\in\chi}|(h_{k,\lambda_\ell})_r|\right\}.\end{aligned} \tag{4.15}$$

Set $H_{k,\ell} = \sum_{r\in\chi} |(h_{k,\lambda_\ell})_r|$. Taking the sum of (4.15) over $s \in \chi$ and picking the maximum over $t \in \chi$, a bound on the ℓ_∞−norm of the Hessian $\nabla^2 J_0$ is obtained:

$$\begin{aligned}\|\nabla^2 J_0(\phi)\|_\infty &= \max_{t\in\chi} \sum_{s\in\chi} \left| \frac{\partial^2 J_0(\phi)}{\partial\phi_t \partial\phi_s} \right| \\ &\leq 4 \sum_{\ell=1}^{3} \sum_{k=1}^{K} \sum_{j\in\chi} \frac{H_{k,\ell}}{\lambda_\ell^2} \left\{ 2 \max_{t\in\chi} |(h_{k,\lambda_\ell})_{j-t}| H_{k,\ell}^2 + \left(H_{k,\ell}^2 + |(o_{k,\lambda_\ell})_j|\right) \left[\max_{t\in\chi} |(h_{k,\lambda_\ell})_{j-t}| + H_{k,\ell} \right] \right\} \\ &= L_0, \quad \forall\ \phi \in \mathbb{R}^{MP}.\end{aligned}$$

From relation $\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty}$ and the fact that $\|\nabla^2 J(\phi)\|_1 = \|\nabla^2 J(\phi)\|_\infty$ ($\nabla^2 J_0(\phi)$ is a symmetric matrix), it follows that $\|\nabla^2 J_0(\phi)\|_2 \leq L_0$ for all $\phi \in \mathbb{R}^{MP}$. Fix $\phi, \psi \in \mathbb{R}^{MP}$. By the mean value theorem for vector-valued functions, we have

$$\|\nabla J_0(\phi) - \nabla J_0(\psi)\|_2 \leq \sup_{\theta\in(0,1)} \|\nabla^2 J_0(\psi + \theta(\phi-\psi))\|_2 \|\phi - \psi\|_2 \leq L_0 \|\phi - \psi\|_2. \tag{4.16}$$

Then ∇J_0 is L_0−Lipschitz continuous and consequently also ∇J is Lipschitz continuous with constant $L = L_0 + L_{TV}$. □

Point (i) of Lemma 4.2 makes clear that, if a solution to problem (4.9) exists, then it is not unique and it can be determined only up to a real constant. This is a common feature shared with the unregularized problem (4.4). However, unlike in (4.4), the objective function J is not periodic and, in addition, none of the two terms J_0 and J_{TV} are coercive, therefore we can not prove the existence of a minimum point of J neither as in Theorem 4.1 nor by coercivity. A specific proof of existence of the solution for problem (4.9) is now presented.

Theorem 4.2. *The objective function J admits at least one global minimum point. Furthermore, if $\psi \in \mathbb{R}^{MP}$ is a global minimizer of J, then also $\{\psi + c\boldsymbol{e} : \ c \in \mathbb{R}\}$ are global minimizers of J.*

Proof. Let $S = \{\phi \in \mathbb{R}^{MP} : \ \phi = c\boldsymbol{e}, \ c \in \mathbb{R}\}$ be the line in $\mathbb{R}^{MP}$ of all constant images and Π any hyperplane intersecting S in one point ϕ_S, i.e.

$$\Pi = \{\phi \in \mathbb{R}^{MP} : \sum_{r\in\chi} a_r \phi_r + b = 0\}, \ \sum_{r\in\chi} a_r \neq 0, \ b \in \mathbb{R}. \tag{4.17}$$

Thanks to part (i) of Lemma 4.2, for any $\phi \in \mathbb{R}^{MP}$ the point $\phi_\Pi = \phi - \left(\frac{\sum_r a_r\phi_r + b}{\sum_r a_r}\right) \boldsymbol{e} \in \Pi$ is such that $J(\phi_\Pi) = J(\phi)$. Consequently, if ψ is a minimum point of J on Π, then it is also a minimum point on $\mathbb{R}^{MP}$, because $J(\psi) \leq J(\phi_\Pi) = J(\phi)$ for all $\phi \in \mathbb{R}^{MP}$. Hence we restrict the search of the minimum point on Π and we denote with $J|_\Pi$ the restriction of J to Π. Since $S = \arg\min_{\phi\in\mathbb{R}^{MP}} J_{TV}(\phi)$ and Π intersects S only in ϕ_S, J_{TV} is a convex function with a unique minimum point on Π, which implies that J_{TV} is coercive on Π. Furthermore, being J_0 periodic and continuous, it is a bounded function on Π. Then $J|_\Pi$ is the sum of a coercive term and a bounded one, therefore it is itself coercive. This allows to conclude that J admits a minimum point on Π and thus also on $\mathbb{R}^{MP}$. The second part of the thesis follows from Lemma 4.2, part (i). □

Note that the above proof of existence holds also for the regularized DIC problem proposed in [115, 116], in which the first-order Tikhonov regularizer used instead of the TV functional is also noncoercive.

4.2 Optimization methods

In previous works [115, 116, 117], the problem of DIC phase reconstruction had been addressed with the nonlinear conjugate gradient method [104]. However, these methods require in practice several evaluations of the objective function and possibly its gradient in order to compute the linesearch parameter. What we propose instead is to tackle problem (4.9) with a gradient descent algorithm in the differentiable case ($\delta > 0$) and a non scaled version of the proximal-gradient method VMILAn, analysed in Chapter 3, in the nondifferentiable case ($\delta = 0$). The key ingredients of both methods are the use of an Armijo linesearch at each iteration, which ensures convergence to a stationary point of problem (4.9), and a clever adaptive choice of the steplength in order to improve the speed of convergence.

For the sake of simplicity, from now on we assume that each monochromatic image is treated as a vector in $\mathbb{R}^N$ (being $N = MP$) obtained by a lexicographic reordering of its pixels.

4.2.1 Gradient and proximal–gradient methods: LMSD and ILA

In this subsection we describe the two proposed algorithms to address problem (4.9) for both cases $\delta > 0$ and $\delta = 0$.

In the former case the problem is unconstrained and differentiable, therefore a gradient descent method can be used. In particular, we exploit the limited memory steepest descent (LMSD) method proposed by Fletcher in [65] and outlined in Algorithm LMSD. The LMSD method is a standard gradient method equipped with a monotone Armijo linesearch and variable steplengths, which are computed as the reciprocals of some suitable approximations of the eigenvalues of the Hessian matrix of the objective function denominated *Ritz values*. This steplength selection rule has already been discussed in Section 1.1.2 and applied in a variety of image processing applications in the previous chapter. Note that the required Ritz values can be practically computed without the explicit knowledge of the Hessian itself, but exploiting only a set of back gradients and steplengths (see steps 6–10 of Algorithm LMSD).

Some practical issues have to be addressed in the implementation of Algorithm LMSD:

- The first loop (step 1 to 5) builds a matrix

$$G = \left[\nabla J(\phi^{(n-m)}) \quad \nabla J(\phi^{(n-m+1)}) \dots \nabla J(\phi^{(n-1)})\right]$$

Algorithm LMSD Limited memory steepest descent (LMSD) method

```
Choose $\rho, \omega \in (0,1)$, $m \in \mathbb{N}_{>0}$, $\alpha_0^{(0)}, \dots, \alpha_{m-1}^{(0)} > 0$, $0 < \alpha_{\min} \le \alpha_{\max}$, $\phi^{(0)} \in \mathbb{R}^N$ and set $n = 0$.
While True
    For $l = 1, \dots, m$
        1. Define $G(:, l) = \nabla J(\phi^{(n)})$.
        2. Compute the smallest non-negative integer $i_n$ such that $\alpha_n = \alpha_n^{(0)} \rho^{i_n}$ satisfies
               $J(\phi^{(n)} - \alpha_n \nabla J(\phi^{(n)})) \le J(\phi^{(n)}) - \omega \alpha_n \|\nabla J(\phi^{(n)})\|^2.$   (4.18)
        3. Compute $\phi^{(n+1)} = \phi^{(n)} - \alpha_n \nabla J(\phi^{(n)})$.
        If "Stopping Criterion" is satisfied
            4. Return
        Else
            5. Set $n = n + 1$.
        EndIf
    EndFor
    6. Define the $(m+1) \times m$ matrix $\Gamma = \begin{bmatrix} \alpha_{n-m}^{-1} & & \\ -\alpha_{n-m}^{-1} & \ddots & \\ & \ddots & \alpha_{n-1}^{-1} \\ & & -\alpha_{n-1}^{-1} \end{bmatrix}$.
    7. Compute the Cholesky factorization $R^T R$ of the $m \times m$ matrix $G^T G$.
    8. Solve the linear system $R^T r = G^T \nabla J(\phi^{(n)})$.
    9. Define the $m \times m$ matrix $\Phi = [R, r] \Gamma R^{-1}$.
    10. Compute the eigenvalues $\theta_1, \dots, \theta_m$ of the symmetric and tridiagonal approximation $\widetilde{\Phi}$ of $\Phi$ defined as
            $\widetilde{\Phi} = \mathrm{diag}(\Phi) + \mathrm{tril}(\Phi, -1) + \mathrm{tril}(\Phi, -1)^T,$
        being diag($\cdot$) and tril($\cdot, -1$) the diagonal and the strictly lower triangular parts of a matrix.
    11. Define $\alpha_{n+i-1}^{(0)} = \max\{\min\{1/\theta_i, \alpha_{\max}\}, \alpha_{\min}\}$, $i = 1, \dots, m$.
EndWhile
```

of size $MP \times m$. The initial values for the first m steplengths can be provided by the user (e.g. by computing the BB ones) or can be chosen with the same approach described in steps 6–10 but with smaller matrices. For example, one can fix $\alpha_0^{(0)}$, compute $G = \nabla J(\phi^{(0)})$ and use steps 6–10 to compute $\alpha_1^{(0)}$. At this point, defining $G = [\nabla J(\phi^{(0)}) \;\; \nabla J(\phi^{(1)})]$ one can compute $\alpha_2^{(0)}$ and $\alpha_3^{(0)}$ and repeat the procedure until a whole set of m back gradients is available.

- The same procedure can be adopted when step 10 provides only $m' < m$ positive eigenvalues. In this case, all columns of G are discarded, G becomes the empty matrix and the algorithm proceeds with m' instead of m until a whole set of m back gradients is computed. If $m' = 0$, a set of m "safeguard" steplengths, corresponding to the last set of m positive steplengths values provided by step 10, is exploited for the next m iterations.
- If $G^T G$ in step 7 is not positive definite, then the oldest gradient of G is discarded and a new matrix $G^T G$ is computed. This step is repeated until $G^T G$ becomes positive definite.
- The stopping criterion can be chosen by the user and be related to the decrease of J or to the distance between two successive iterates. In our tests we decided to arrest the iterations when the norm of the gradient ∇J goes below a given threshold κ:

$$\|\nabla J(\phi^{(n)})\| \le \kappa. \tag{4.19}$$

Concerning the computational costs of LMSD, the heaviest tasks at each iteration are the computation of $\nabla J(\phi^{(n)})$ at step 1 and $J(\phi^{(n)} - \alpha_n \nabla J(\phi^{(n)}))$ at step 2. Considering step 1, we focus on ∇J_0. As it is written in (4.12), due to the product between $e^{-i\phi_s/\lambda_\ell}$ and $(h_{k,\lambda_\ell})_{j-s}$, ∇J_0 can be performed with $\mathcal{O}(N^2)$ complexity; this is how the gradient is computed in [115]. However, if we take the sum over j of the residuals into the argument of $\mathrm{Im}(\cdot)$, then we can conveniently rewrite (4.12) as

$$\frac{\partial J_0(\phi)}{\partial \phi_s} = \sum_{\ell=1}^{3}\sum_{k=1}^{K} \frac{4}{\lambda_\ell} \mathrm{Im}\Big\{ \Big((r_{k,\lambda_\ell} .* (\overline{h_{k,\lambda_\ell}} \otimes e^{i\phi/\lambda_\ell})) \otimes \tilde{h}_{k,\lambda_\ell} \Big)_s e^{-i\phi_s/\lambda_\ell} \Big\}, \tag{4.20}$$

where $h_1 .* h_2$ denotes the componentwise product between two images h_1, h_2 and $(\tilde{h}_{k,\lambda_\ell})_j = (h_{k,\lambda_\ell})_{-j}$ for all $j \in \chi$. Then the heaviest operations in (4.20) are the two convolutions which, thanks to the assumption of periodic boundary conditions, can be performed with a FFT/IFFT pair ($\mathcal{O}(N \log N)$ complexity). Hence, since ∇J_{TV} has $\mathcal{O}(N)$ complexity, we can conclude that step 1 has an overall complexity of $\mathcal{O}(N \log N)$. Similarly, the function at step 2 is computed with complexity $\mathcal{O}(N \log N)$, due to the presence of one convolution inside the triple sum in (4.3).

From a practical point of view, the LMSD method has proven to be an effective tool for DIC imaging, especially if compared to more standard gradient methods equipped with the BB rules [13]. From a theoretical point of view, let us remark that LMSD can be seen as a special

```
Algorithm ILA Inexact Linesearch based Algorithm (ILA)
Choose $0 < \alpha_{\min} \leq \alpha_{\max}$, $\rho, \omega \in (0,1)$, $\gamma \in [0,1]$, $\tau > 0$, $\phi^{(0)} \in \mathbb{R}^N$ and set $n = 0$.
While True
  1. Set $\alpha_n = \max\left\{\min\left\{\alpha_n^{(0)}, \alpha_{\max}\right\}, \alpha_{\min}\right\}$, where $\alpha_n^{(0)}$ is chosen as in Algorithm LMSD.
  2. Let $\psi^{(n)} = \arg\min_{\phi \in \mathbb{R}^n} h^{(n)}(\phi) = \text{prox}_{\alpha_k J_{TV}}\left(\phi^{(n)} - \alpha_k \nabla J_0(\phi^{(n)})\right)$.
     Compute $\tilde{\psi}^{(n)}$ such that
         $h^{(n)}(\tilde{\psi}^{(n)}) - h^{(n)}(\psi^{(n)}) \leq -\frac{\tau}{2} h_\gamma^{(n)}(\tilde{\psi}^{(n)}).$   (4.21)
     where $h_\gamma^{(n)}(\tilde{\psi}^{(n)}) = \nabla J_0(\phi^{(n)})^T(\tilde{\psi}^{(n)} - \phi^{(n)}) + \frac{\gamma}{2\alpha_k}\|\tilde{\psi}^{(n)} - \phi^{(n)}\|^2 + J_{TV}(\tilde{\psi}^{(n)}) - J_{TV}(\phi^{(n)})$.
  3. Set $d^{(n)} = \tilde{\psi}^{(n)} - \phi^{(n)}$.
  4. Compute the smallest non-negative integer $i_n$ such that $\lambda_n = \rho^{i_n}$ satisfies
         $J(\phi^{(n)} + \lambda_n d^{(n)}) \leq J(\phi^{(n)}) + \omega \lambda_n h_\gamma^{(n)}(\tilde{\psi}^{(n)}).$   (4.22)
  5. Compute the new point as $\phi^{(n+1)} = \phi^{(n)} + \lambda_n d^{(n)}$.
     If "Stopping Criterion" is satisfied
       6. Return
     Else
       7. Set $n = n + 1$.
     EndIf
EndWhile
```

instance of Algorithm VMILAn in Chapter 3, if we set $f_1 = \iota_{\mathbb{R}^n}$, $D_k = I_n$ and $\gamma = 0$. This allows to state that the sequence $\{\phi^{(n)}\}_{n \in \mathbb{N}}$ converges to a limit point (if any exists) which is stationary for J. In fact, ∇J is Lipschitz continuous (Lemma 4.2) and, in addition, J also satisfies the Kurdyka–Łojasiewicz (KL) property. Indeed J_0 is an analytic function (Lemma 4.1, part (iii)) and J_{TV} is a semialgebraic function, which means that its graph is defined by a finite sequence of polynomial equations and inequalities (see Definition 2.15 and Example 2.18 in Chapter 2). Since J is the sum of an analytic function and a semialgebraic one, this is sufficient to conclude that it satisfies the KL property on $\mathbb{R}^N$. At this point, we invoke Corollary 3.1, which states the convergence of the method.

We now turn to the nonsmooth case $\delta = 0$, which has been addressed by means of a simplified version of VMILAn. In its general form, VMILAn exploits a variable metric in the (possibly inexact) computation of the proximal point at each iteration, and a backtracking loop to satisfy an Armijo–like inequality. Effective variable metrics can be designed for specific objective functions either by exploiting the splitting gradient idea or the majorize-minimize technique (see Section 3.3). However, since in the DIC problem the gradient of J_0 does not lead to a natural decomposition in the required form, in our tests we used the standard Euclidean distance (we will denote with ILA this simplified version of VMILAn).

The main steps of ILA are detailed in Algorithm ILA. We recall that, at each iteration n, given the point $\phi^{(n)} \in \mathbb{R}^N$ and the parameters $\alpha_n > 0$, $\gamma \in [0,1]$, the metric function $h_\gamma^{(n)}$ is defined as

$$h_\gamma^{(n)}(\phi) = \nabla J_0(\phi^{(n)})^T(\phi - \phi^{(n)}) + \frac{\gamma}{2\alpha_n}\|\phi - \phi^{(n)}\|^2 + J_{TV}(\phi) - J_{TV}(\phi^{(n)}). \tag{4.23}$$

By setting $h^{(n)} = h_1^{(n)}$ and $z^{(n)} = \phi^{(n)} - \alpha_n \nabla J_0(\phi^{(n)})$, the proximal-gradient point is then computed as

$$\psi^{(n)} := \mathrm{prox}_{\alpha_n J_{TV}}(z^{(n)}) = \arg\min_{\phi\in\mathbb{R}^N} h^{(n)}(\phi). \tag{4.24}$$

In step 2 of Algorithm ILA, an approximation $\tilde{\psi}^{(n)}$ of the proximal point $\psi^{(n)}$ is defined by means of condition (4.21). As already seen in Section 3.2.4, such a point can be practically computed by remarking that J_{TV} can be written as

$$J_{TV}(\phi) = g(\mathcal{D}\phi), \qquad g(t) = \mu \sum_{j=1}^{N} \left\| \begin{pmatrix} t_{2j-1} \\ t_{2j} \end{pmatrix} \right\|, \qquad t \in \mathbb{R}^{2N}.$$

Then considering the dual problem of (4.24)

$$\max_{v\in\mathbb{R}^{2N}} \Gamma^{(n)}(v), \tag{4.25}$$

the dual function $\Gamma^{(n)}$ has the following form

$$\Gamma^{(n)}(v) = -\frac{\|\alpha_n \mathcal{D}^T v - z^{(n)}\|^2}{2\alpha_n} - g^*(v) - J_{TV}(\phi^{(n)}) - \frac{\alpha_n}{2}\|\nabla J_0(\phi^{(n)})\|^2 + \frac{\|z^{(n)}\|^2}{2\alpha_n} \tag{4.26}$$

where the convex conjugate g^* is the indicator function of the set $\left(B^2(0,\mu)\right)^N$, being $B^2(0,\mu) \subset \mathbb{R}^2$ the 2-dimensional Euclidean ball centered in 0 with radius μ.
Condition (4.21) is fulfilled by any point $\tilde{\psi}^{(n)} = z^{(n)} - \alpha_n A^T v^{(n)}$ with $v^{(n)} \in \mathbb{R}^{2N}$ satisfying

$$h^{(n)}(\tilde{\psi}^{(n)}) \leq \eta \Gamma^{(n)}(v^{(n)}), \qquad \eta = 2/(2+\tau). \tag{4.27}$$

If an iterative method is applied to the dual problem, generating a sequence $\{v^{(n,\ell)}\}_{\ell\in\mathbb{N}}$ such that convergence is guaranteed for both the iterates and function values, and setting $\tilde{\psi}^{(n,\ell)} =$

$z^{(n)} - \alpha_n A^T v^{(n,\ell)}$ for all ℓ, then (4.27) will be satisfied for all sufficiently large ℓ.

Since the gradient of f_0 is Lipschitz continuous, Theorem 3.2 ensures that each limit point of the ILA sequence is stationary.

4.2.2 Nonlinear conjugate gradient methods

We compare the performances of LMSD and ILA with several nonlinear conjugate gradient methods, including some standard CG methods [104, 63] and the heuristic CG method previously used for DIC problems [117, 115]. The general scheme for a CG method is recalled in Algorithm CG and some classical choices for the parameter β_{n+1} are shown in Table 4.1, namely the Fletcher-Reeves (FR), Polak-Ribière (PR), PR with nonnegative values (PR$^+$) and PR constrained by the FR values (FR-PR) strategies [71].

Algorithm CG Conjugate gradient (CG) method

```
Choose $\phi^{(0)} \in \mathbb{R}^N$ and set $n = 0$, $p^{(0)} = -\nabla J(\phi^{(0)})$.
While True
    1. Compute $\alpha_n$ and set $\phi^{(n+1)} = \phi^{(n)} + \alpha_n p^{(n)}$.
    2. Choose the scalar parameter $\beta_{n+1}$ according to the CG strategy used.
    3. Define $p^{(n+1)} = -\nabla J(\phi^{(n+1)}) + \beta_{n+1} p^{(n)}$.
    If "Stopping Criterion" is satisfied
        4. Return
    Else
        5. Set $n = n + 1$.
    EndIf
EndWhile
```

In order to ensure the global convergence of the FR and FR-PR methods, the steplength parameter α_n in step 1 must comply with the strong Wolfe conditions [71, 104]

$$\begin{aligned} J(\phi^{(n)} + \alpha_n p^{(n)}) &\leq J(\phi^{(n)}) + c_1 \alpha_n \nabla J(\phi^{(n)})^T p^{(n)} \\ |\nabla J(\phi^{(n)} + \alpha_n p^{(n)})^T p^{(n)}| &\leq c_2 |\nabla J(\phi^{(n)})^T p^{(n)}| \end{aligned} \tag{4.28}$$

where $0 < c_1 < c_2 < \frac{1}{2}$. Concerning the PR methods, one can prove convergence if β_{n+1} is chosen according to the PR$^+$ rule and α_n satisfies both (4.28) and the following additional

CG algorithm	β_{n+1}
FR	$\beta_{n+1}^{\mathrm{FR}} = \dfrac{\nabla J(\phi^{(n+1)})^T \nabla J(\phi^{(n+1)})}{\nabla J(\phi^{(n)})^T \nabla J(\phi^{(n)})}$
PR	$\beta_{n+1}^{\mathrm{PR}} = \dfrac{\nabla J(\phi^{(n+1)})^T (\nabla J(\phi^{(n+1)}) - \nabla J(\phi^{(n)}))}{\nabla J(\phi^{(n)})^T \nabla J(\phi^{(n)})}$
PR$^+$	$\beta_{n+1}^{\mathrm{PR}^+} = \max(\beta_{n+1}^{\mathrm{PR}}, 0)$
FR-PR	$\beta_{n+1}^{\mathrm{FRPR}} = \begin{cases} \beta_{n+1}^{\mathrm{PR}} & if \lvert\beta_{n+1}^{\mathrm{PR}}\rvert \le \beta_{n+1}^{\mathrm{FR}} \\ \beta_{n+1}^{\mathrm{FR}} & otherwise \end{cases}$

Table 4.1: Choice of the parameter β_{n+1} in CG methods. From top to bottom: Fletcher-Reeves (FR), Polak-Ribière (PR), Polak-Ribière with nonnegative β_{n+1} (PR$^+$), Polak-Ribière constrained by the FR method (FR-PR).

condition [71, 104]

$$\nabla J(\phi^{(n)})^T p^{(n)} \le -c_3 \|\nabla J(\phi^{(n)})\|^2, \qquad 0 < c_3 \le 1. \tag{4.29}$$

For a practical implementation of a backtracking method to satisfy (4.28) see e.g. [104, Section 3.5], while for the addition of condition (4.29) see [71, Section 6]. In Section 4.3, the CG methods equipped with the FR, FR-PR, PR$^+$ rules for the parameter β_{n+1}, together with conditions (4.28) for the linesearch parameter α_n, will be denominated FR-SW, FR-PR-SW and PR$^+$-SW respectively, where SW stands for Strong Wolfe conditions.

Since in the DIC problem the evaluation of the gradient ∇J is computational demanding and its nonlinearity w.r.t. α requires a new computation for each step of the backtracking loop, in [117, 115] a heuristic version of the FR and PR methods is used exploiting a linesearch based on a polynomial approximation method. The resulting scheme for the choice of α_n is detailed in Algorithm PA, even if we recognize that our routines might differ from those used in [117, 115] due to the lack of several details crucial for reproducing their practical implementation. As we will see in the next Section, this linesearch is quite sensitive to the choice of the parameter t. Moreover, since the strong Wolfe conditions are not imposed, there is no guarantee that the FR or PR methods endowed with this choice for α_n converges, nor that $p^{(n+1)}$ is a descent direction for all n. In the following, the CG methods equipped with the FR and PR rule, together with the linesearch described in Algorithm PA, will be indicated as FR-PA and PR-PA respectively, where PA stands for polynomial approximation.

4.3 Numerical experiments

In this section we test the effectiveness of the algorithms previously described in some synthetic problems. All the numerical results have been obtained on a PC equipped with an INTEL Core

Algorithm PA Linesearch based on polynomial approximation

```
Let $\psi(\alpha) := J(\phi^{(n)} + \alpha p^{(n)})$ and set $t > 0$, $a = 0$, $b = t$.
Compute $\psi(a)$ and $\psi(b)$.
  1. Find a point $c \in [a, b]$ such that $\psi(a) > \psi(c) < \psi(b)$ as follows
     If $\psi(b) < \psi(a)$
         Set $c = 2b$ and compute $\psi(c)$.
         While $\psi(c) \leq \psi(b)$
             Set $a = b$, $b = c$, $c = 2c$ and compute $\psi(c)$.
         EndWhile
     Else
         Set $c = \frac{b}{2}$ and compute $\psi(c)$.
         While $\psi(c) \geq \psi(a)$
             Set $b = c$, $c = \frac{c}{2}$ and compute $\psi(c)$.
         EndWhile
     EndIf
  2. Compute $\alpha_n$ as the minimum point of the parabola interpolating the points
     $(a, \psi(a)), (b, \psi(b)), (c, \psi(c))$.
```

i7 processor 2.60GHz with 8GB of RAM running Matlab R2013a with its standard settings. The LMSD and ILA routines for the DIC problem together with an illustrative example can be downloaded from [124].

4.3.1 Comparison between LMSD and CG methods

The evaluations of the various optimization methods discussed in Section 4.2 have been carried out on two phantom objects (see Figure 4.5), which have been computed by using the formula for the phase difference between two waves travelling through two different media

$$\phi_s = 2\pi(n_1 - n_2)t_s, \tag{4.30}$$

where n_1 and n_2 are the refractive indices of the object structure and the surrounding medium, respectively, and t_s is the thickness of the object at pixel $s \in \chi$. The first phantom, denominated "cone" and reported at the top row of Figure 4.5, is a 64×64 phase function representing a truncated cone of radius $r = 3.2$ μm with $n_1 = 1.33$, $n_2 = 1$ and maximum value $\phi_{\max} =$

1.57 rad attained at the cone vertex. The “cross” phantom, shown at the bottom row of Figure 4.5, is another 64×64 phase function of two crossing bars, each one of width 5 μm, measuring 0.114 rad inside the bars and 0 in the background. For both simulations, the DIC microscope parameters were set as follows:

- shear: $2\Delta x = 0.6$ μm;
- bias: $2\Delta\theta = \pi/2$ rad;
- numerical aperture of the objective: NA $= 0.9$.

For each phantom, a dataset consisting of $K = 2$ polychromatic DIC images acquired at shear angles $\tau_1 = -\pi/4$ rad and $\tau_2 = \pi/4$ rad was created, as in model (4.1), by convolving the true phase function with the accordingly rotated DIC PSFs and then by corrupting the result with white Gaussian noise at different values of the signal-to-noise ratio

$$\mathrm{SNR} = 10 \log_{10} \left(\frac{\overline{\phi^*}}{\sigma} \right) \tag{4.31}$$

where $\overline{\phi^*}$ is the mean value of the true object and σ is the standard deviation of noise. The SNR values chosen in the simulations were 9 dB and 4.5 dB.

As far as the regularization parameter μ and the threshold δ in (4.10) are concerned, these have been manually chosen from a fixed range in order to obtain a visually satisfactory reconstruction. Note that the parameters were first set in the differentiable case ($\delta > 0$) for the LMSD and the nonlinear CG methods and then the same value of the parameter μ was used also in the nondifferentiable case ($\delta = 0$) for the ILA method. The values reported below have been used for each simulation presented in this section. The resulting values have been $\mu = 10^{-2}, \delta = 10^{-2}$ for the cone and $\mu = 4 \cdot 10^{-2}, \delta = 10^{-3}$ for the cross.

Some details regarding the choice of the parameters involved in the optimization methods of Section 4.2 are now provided. The linesearch parameters ρ, ω of the LMSD and ILA methods have been respectively set to 0.5, 10^{-4}. These are the standard choices for the Armijo parameters, however it is known that the linesearch algorithm is not so sensible to modifications of these values [37, 112]. The parameter γ in the Armijo–like rule (3.24) has been fixed equal to 1, which corresponds to the mildest choice in terms of decrease of the objective function J. The parameter m in Algorithm LMSD is typically a small value ($m = 3, 4, 5$), in order to avoid a significant computational cost in the calculation of the steplengths $\alpha_n^{(0)}$; here we let $m = 4$. The same choice for m is done in Algorithm ILA, where the values $\alpha_n^{(0)}$ are constrained in the interval $[\alpha_{\min}, \alpha_{\max}]$ with $\alpha_{\min} = 10^{-5}$ and $\alpha_{\max} = 10^2$. As done in the experiments of the previous chapter, the dual problem (4.25) is addressed, at each iteration of ILA, by means of algorithm FISTA [14] which is stopped by using criterion (3.72) with $\eta = 10^{-6}$. This value

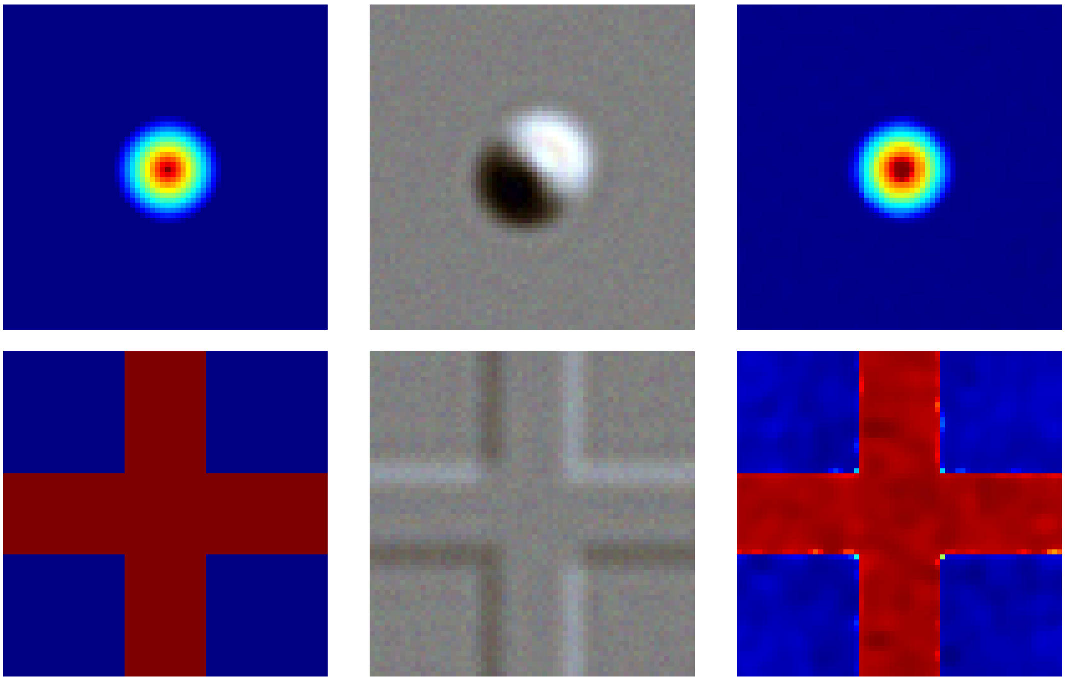


Figure 4.5: Data and results for the cone (top row) and cross (bottom row) objects. From left to right: true object, noisy DIC color image taken at shear angle $\frac{\pi}{4}$ rad and corrupted with white Gaussian noise at SNR = 4.5 dB, and reconstructed phase with the LMSD method from observations at shear angles equal to $-\pi/4$ rad and $\pi/4$ rad.

represents a good balance between convergence speed and computational time per iteration [32]. Concerning the nonlinear CG methods equipped with the strong Wolfe conditions, we set $c_1 = 10^{-4}$ and $c_2 = 0.1$ in (4.28) as done in [71] and we initialize the related backtracking procedure as suggested in [104, p. 59]. Regarding the CG methods endowed with the polynomial approximation detailed in Algorithm PA, a restart of the method is performed by setting $\beta_{n+1} = 0$, hence by taking a steepest descent step, whenever the vector $p^{(n+1)}$ fails to be a descent direction. Finally, the constant phase object $\phi^{(0)} = 0$ is chosen as initial guess for all methods.

In order to evaluate the performance of the phase reconstruction methods proposed in Section 4.2, we will make use of the following error distance

$$E(\phi^{(n)}, \phi^*) = \min_{c\in\mathbb{R}} \frac{\|\phi^{(n)} - \phi^* - c\boldsymbol{e}\|}{\|\phi^*\|} = \frac{\|\phi^{(n)} - \phi^* - \bar{c}\boldsymbol{e}\|}{\|\phi^*\|} \tag{4.32}$$

where ϕ^* is the phase to be reconstructed and $\bar{c} = \sum\limits_{j\in\chi} \frac{(\phi_j^{(n)} - \phi_j^*)}{N}$. Unlike the usual root mean squared error, which is recovered by setting $c = 0$ in (4.32), the error distance defined in (4.32)

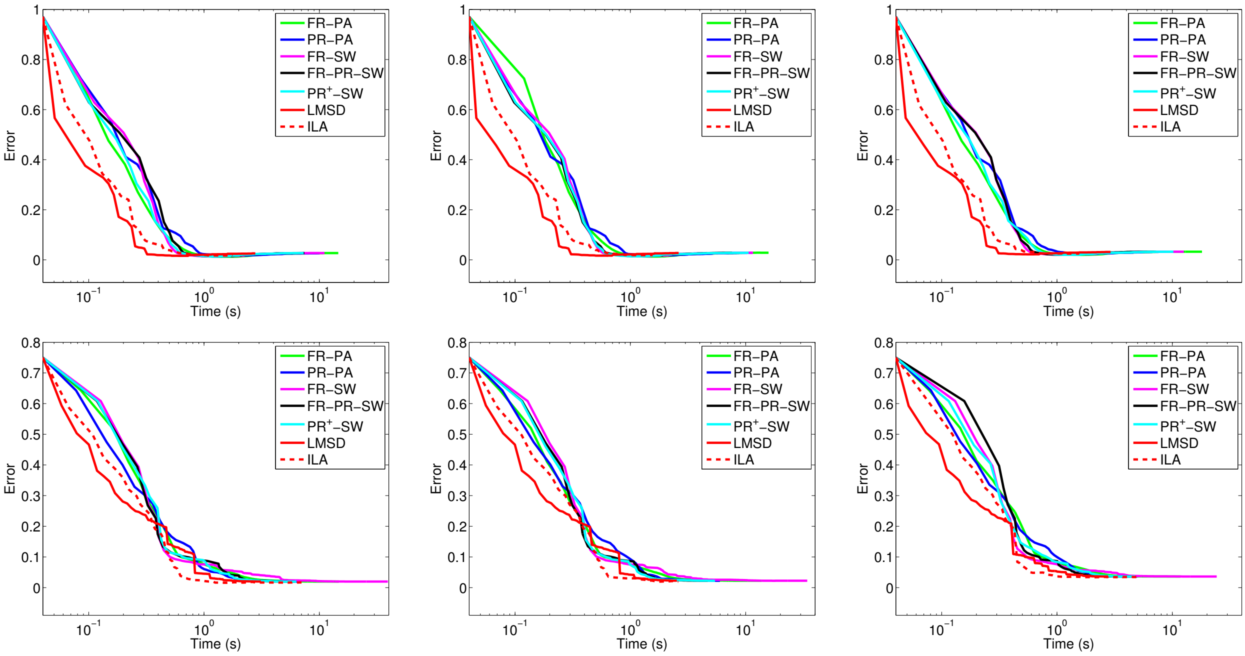


Figure 4.6: Error versus computational time plots for the cone (top row) and cross (bottom row) objects. From left to right: noise-free data, SNR = 9 dB and SNR = 4.5 dB.

is invariant with respect to phase shifts, i.e.

$$E(\phi + c\boldsymbol{e}, \phi^*) = E(\phi, \phi^*), \quad \forall \phi \in \mathbb{R}^N, \ \forall c \in \mathbb{R}. \tag{4.33}$$

That makes the choice of (4.32) well-suited for problem (4.9), whose solution might be recovered only up to a real constant.

The methods have been run for the cone and cross phantoms with the parameter setting outlined in the previous subsection. The iterations of the LMSD and the CG methods have been arrested when the stopping criterion (4.19) was met with $\kappa = 10^{-3}$, while the ILA method has been stopped when the error up-to-a-constant between two successive iterates was lower than a prefixed $\kappa > 0$, that is

$$\frac{\left\| \phi^{(n+1)} - \phi^{(n)} - \left(\overline{\phi^{(n+1)} - \phi^{(n)}} \right) \boldsymbol{e} \right\|}{\|\phi^{(n+1)}\|} \leq \kappa, \tag{4.34}$$

where $\overline{\phi^{(n+1)} - \phi^{(n)}}$ is the mean value of the difference between the two objects. The tolerance κ in (4.34) was set equal to $5 \cdot 10^{-5}$ for the cone and 10^{-4} for the cross.

In Figure 4.6 we show the reconstruction error provided by the different methods as a function of the computational time. We start by comparing LMSD with the CG methods equipped with Algorithm PA (FR-PA, PR-PA) and the CG methods equipped with the Strong Wolfe conditions (FR-SW, FR-PR-SW, PR$^+$-SW). From the plots of Figure 4.6, it can be drawn that each method is quite stable with respect to the noise level on the DIC images. However, in terms of time efficiency, LMSD outperforms all the CG methods in the cone tests, showing a time reduction of nearly 50% to achieve the smallest error. Furthermore, what emerges by looking at Tables 4.2 and 4.3 is that the CG methods are much more computationally demanding than LMSD. For instance, in the case of the cone (Table 4.1), LMSD evaluates the function on average less than 2 times per iteration. By contrast, the backtracking procedure exploited in the FR, FR-PR and PR$^+$ methods requires an average of $4-5$ evaluations per iteration of both the function and gradient to satisfy the strong Wolfe conditions, whereas the FR-PA and PR-PA methods, despite evaluating the gradient only once, need on average $10-12$ evaluations of the function before detecting the three-points-interval described in Algorithm PA. One could reduce the number of evaluations in FR-PA and PR-PA by properly tuning the parameter t in Algorithm PA. However, as it is evident from Table 4.4, these methods are quite sensitive to the choice of t, as little variations of this parameter might result in a great increase of the number of restarts and, eventually, in the divergence of the algorithm. In addition, it seems that the optimal value of t strictly depends on the object to be reconstructed.

4.3.2 Comparison between LMSD and ILA

We now compare the performances of LMSD and ILA. On one hand, ILA reconstructs the cross object slightly better than LMSD. Indeed, ILA provides the lowest reconstruction error in Table 4.3 for each SNR value and the corresponding phase estimates have better preserved edges, as clearly depicted in Figure 4.7, where we consider the following "up-to-a-constant" residual

$$R_j = \left|\phi_j - \phi_j^* - \overline{\phi - \phi^*}\right|, \quad \forall j \in \chi \tag{4.35}$$

to measure the quality of the reconstructions provided by the two methods. This result was expected, since ILA addresses problem (4.9) with the standard TV functional ($\delta = 0$ in (4.10)), which is more suited than HS regularization ($\delta > 0$) when the object to be reconstructed is piecewise-constant. On the other hand, ILA may be computationally more expensive since, unlike LMSD, it requires to iteratively solve the inner subproblem (4.25) at each outer iteration. Indeed, looking at Table 4.3 we notice that, although the number of function evaluations per iteration in LMSD and ILA is quite similar (on average around 1.4 for LMSD and 1.8 for ILA) and the ILA iterations are stopped way before the LMSD ones, the computational time in ILA is always higher. For instance, in the case SNR = 9 dB, the methods require approximately the same time, although the number of iterations of ILA is more than halved. This fact is explained if we look at the average number of inner iterations required by ILA to compute the approximate proximal point: 21.3, 10.11 and 13.43 for SNR $= \infty, 9, 4.5$ dB respectively. Analogous conclusions can be drawn by considering the results on the cone object (see Table

SNR (dB)	Algorithm	Iterations	# f	# g	Time (s)	Obj fun	Error
∞	FR–PA	280	3016	280	14.60	0.89	2.72 %
	PR–PA	168	2137	168	10.10	0.89	2.66 %
	FR–SW	183	770	770	11.08	0.89	2.73 %
	FR-PR–SW	127	514	514	7.41	0.89	2.71 %
	PR^+–SW	129	504	504	7.32	0.89	2.71 %
	LMSD	153	212	153	2.77	0.89	2.60 %
	ILA	66	119	66	1.77	0.52	1.76 %
9	FR–PA	306	3245	306	15.79	1.65	2.85 %
	PR–PA	188	2393	188	11.41	1.65	2.80 %
	FR–SW	194	804	804	11.60	1.65	2.85 %
	FR-PR–SW	134	520	520	7.61	1.65	2.84 %
	PR^+–SW	144	734	734	10.61	1.65	2.84 %
	LMSD	149	197	149	2.61	1.65	2.75 %
	ILA	60	91	60	1.56	1.29	1.91 %
4.5	FR–PA	347	3696	347	18.08	6.88	3.26 %
	PR–PA	146	1858	146	8.84	6.88	3.24 %
	FR–SW	204	867	867	12.58	6.88	3.26 %
	FR-PR–SW	152	492	492	7.24	6.88	3.26 %
	PR^+–SW	144	701	701	10.22	6.88	3.26 %
	LMSD	163	228	163	2.90	6.88	3.17 %
	ILA	61	104	61	1.56	6.80	2.50 %

Table 4.2: Cone tests. From left to right: number of iterations required to meet the stopping criteria, number of function and gradient evaluations, execution time, objective function value and error achieved at the last iteration.

4.2).

In order to deepen the analysis between the differentiable TV approximation and the original nondifferentiable one, we compared the LMSD and ILA methods in one further realistic simulation. In particular, we considered the “grid” object in Figure 4.8, which is a 1388×1040 image emulating the phase function of a multi-area calibration artifact [119, 125], which measures 1.212 rad inside the black regions and 2.187 rad inside the white ones. The setup of the two methods is identical to that of the previous tests (with the exception of the numerical aperture of the objective NA which has been set equal to 0.8), and the parameters μ (for both models) and δ (for the smooth TV functional) have been set equal to $2 \cdot 10^{-1}$ and 10^{-1}, respectively. Instead of three levels of noise, here we only considered a SNR equal to 9 dB. In Figure 4.9 we report the behaviour of the error (4.32) as a function of time and the number of inner iterations needed by ILA to address problem (4.25)–(3.72).

SNR (dB)	Algorithm	Iterations	# f	# g	Time (s)	Obj fun	Error
∞	FR–PA	412	2618	412	14.75	1.01	1.98 %
	PR–PA	138	1373	138	6.73	1.01	1.98 %
	FR–SW	411	2768	2768	39.27	1.01	1.98 %
	FR-PR–SW	109	423	423	6.14	1.01	1.98 %
	PR$^+$–SW	116	438	438	6.32	1.01	1.98 %
	LMSD	168	231	168	3.09	1.01	2.00 %
	ILA	100	176	100	7.18	0.87	1.66 %
9	FR–PA	391	2490	391	13.77	1.96	2.25 %
	PR–PA	121	1209	121	5.97	1.96	2.26 %
	FR–SW	388	2417	2417	34.18	1.96	2.25 %
	FR-PR–SW	106	323	323	4.69	1.96	2.25 %
	PR$^+$–SW	109	375	375	5.41	1.96	2.25 %
	LMSD	140	190	140	2.52	1.96	2.27 %
	ILA	57	106	57	2.60	1.82	1.94 %
4.5	FR–PA	303	2164	303	11.74	8.57	3.63 %
	PR–PA	98	997	98	4.97	8.57	3.63 %
	FR–SW	299	1705	1705	24.28	8.57	3.63 %
	FR-PR–SW	96	300	300	4.41	8.57	3.63 %
	PR$^+$–SW	98	326	326	4.74	8.57	3.63 %
	LMSD	152	221	152	2.75	8.57	3.64 %
	ILA	97	179	97	5.26	8.47	3.46 %

Table 4.3: Cross tests. From left to right: number of iterations required to meet the stopping criteria, number of function and gradient evaluations, execution time, objective function value and error achieved at the last iteration.

The grid dataset confirms the remarks previously done, since ILA takes almost twice the time than LMSD to provide an estimate of the phase. This is again due to the number of inner iterations, which starts to oscillatory increase after the first 20 iterations (see Figure 4.9). To conclude, we reckon that the LMSD method is generally preferable since, unlike ILA, it does not require any inner subproblem to be solved and thus it is generally less expensive from the computational point of view. However, the ILA method should be considered as a valid alternative when the object to be reconstructed is piecewise-constant.

Dataset	t	Iterations	# f	Time (s)	Obj fun	Error	Restarts
	10^{-4}	500	4272	24.57	8.57	3.63 %	8
Cross	10^{-3}	500	2911	19.66	8.57	3.63 %	6
-	$5 \cdot 10^{-3}$	500	3073	19.63	8.57	3.63 %	1
SNR	10^{-2}	500	5337	28.97	8.57	3.63 %	21
4.5 dB	$5 \cdot 10^{-2}$	500	2023	15.22	8.59	3.91 %	424
	10^{-1}	500	2032	15.44	8.88	5.05 %	365
	10^{-3}	500	4788	26.13	6.88	3.27 %	0
Cone	10^{-2}	500	3260	19.84	6.88	3.27 %	0
-	10^{-1}	500	2126	15.86	6.88	3.27 %	3
SNR	$2 \cdot 10^{-1}$	500	2427	16.78	6.88	3.27 %	0
4.5 dB	$2.25 \cdot 10^{-1}$	500	1610	13.39	1507.4	130.94 %	41
	$2.5 \cdot 10^{-1}$	500	1713	13.67	2373.4	315.50 %	87

Table 4.4: Setting the parameter t in the PR-PA algorithm.

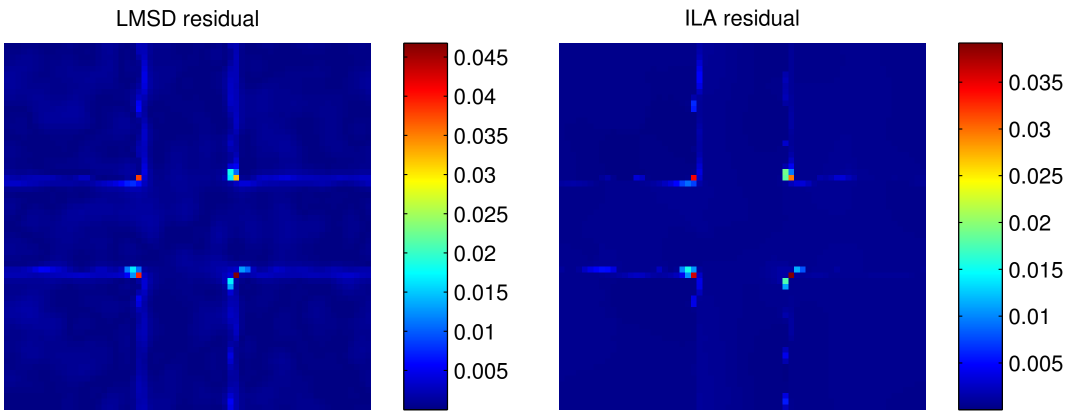


Figure 4.7: Cross test. The residuals defined in (4.35) for the reconstructions provided by LMSD and ILA, respectively, when the acquired images are corrupted with SNR = 9 dB.

4.3.3 Influence of color and bias retardation on phase reconstruction

Another analysis of our interest was to observe how color information and bias retardation in the observations affect the behavior of phase reconstruction. We set four scenarios for comparison: independent monochromatic observations with red, green, and blue light, and polychromatic observation where all wavelengths are combined. For each of these scenarios we used the cross object to generate 100 observations at different realizations of noise, for both SNR = 4.5 dB and SNR = 9 dB, and bias retardation of 0 rad and $\pi/2$ rad, at shear angles equal to $-\pi/4$ rad and $\pi/4$ rad. We tested the LMSD method to perform the reconstructions; results for SNR =

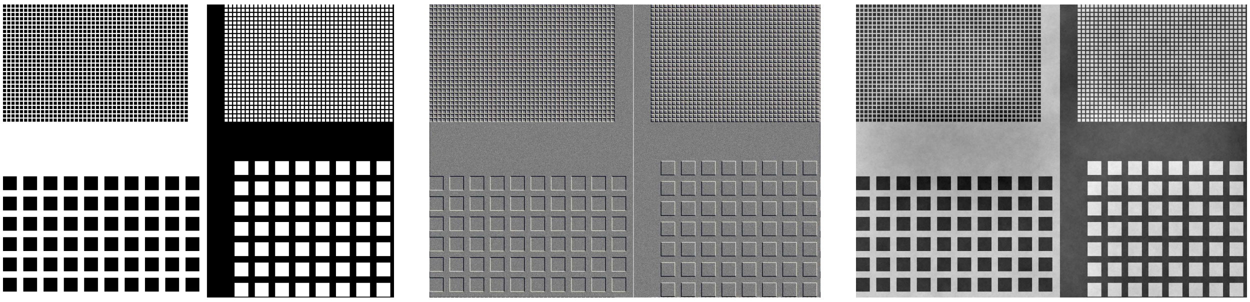


Figure 4.8: Data and results for the grid object. From left to right: true object, noisy DIC color image taken at shear angle $\frac{\pi}{4}$ rad and corrupted with white Gaussian noise at SNR = 9 dB, and reconstructed phase with the LMSD method from observations at shear angles equal to $-\pi/4$ rad and $\pi/4$ rad.

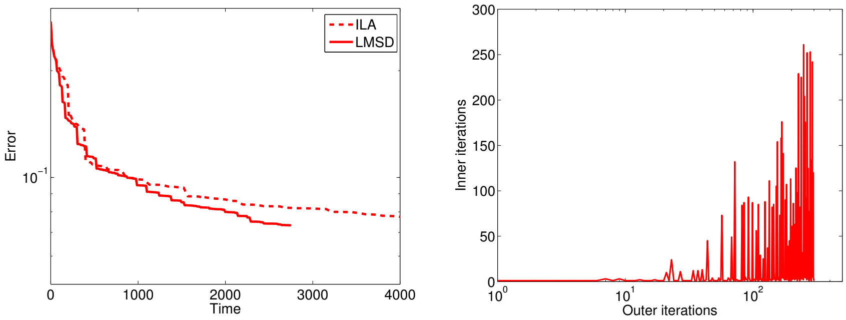


Figure 4.9: Grid test. From left to right: error versus time plots for LMSD and ILA and number of inner iterations versus number of outer iterations for ILA.

4.5 dB are shown in Figure 4.10 and for SNR = 9 dB in Figure 4.11.

The lines show the average error over the 100 observations. It is noticed that for 0 rad bias retardation, the reconstruction for polychromatic observations behave better than for the monochromatic ones, even though the amount of error is not promising of a good reconstruction. For $\pi/2$ rad bias retardation the algorithm stops before the maximum number of iterations (500) is reached. In this case, for both levels of noise, the performance of the reconstruction with polychromatic light is quite comparable with monochromatic light. Another interesting finding about the convergence for monochromatic light, is that for all cases, it happens in the order red-green-blue; this is due to the fact that the amplitude PSF for blue light has the bigger frequency support, thus provides more information for reconstruction.

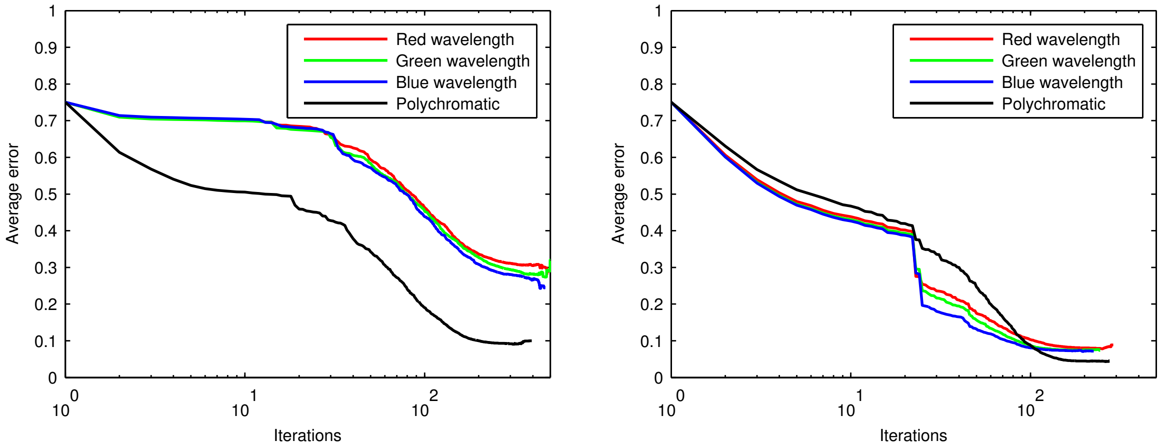


Figure 4.10: Average error comparison between monochromatic and polychromatic reconstructions. SNR = 4.5 dB. Left: bias 0 rad; right: bias $\pi/2$ rad.

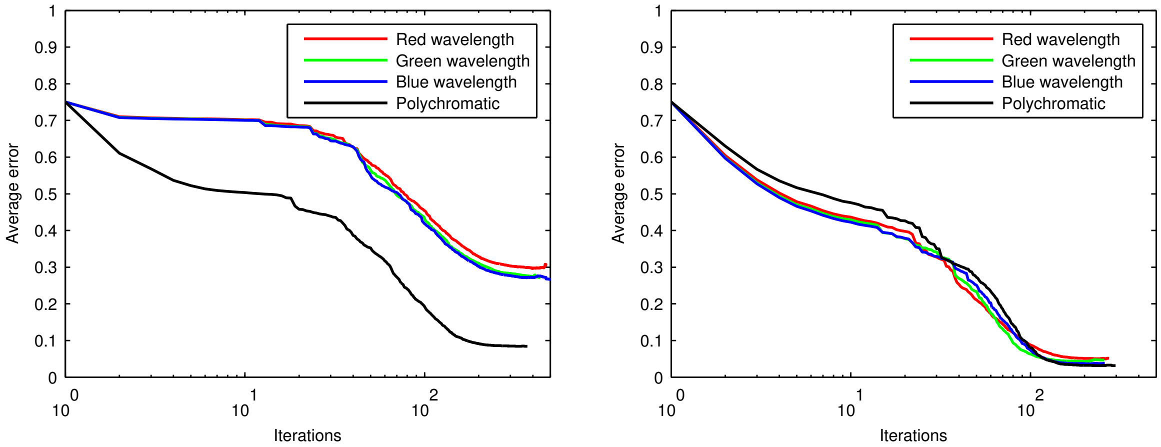


Figure 4.11: Average error comparison between monochromatic and polychromatic reconstructions. SNR = 9 dB. Left: bias 0 rad; right: bias $\pi/2$ rad.

Chapter 5

A cyclic block generalized gradient projection method

This chapter deals with the following optimization problem

$$\min_{x\in\Omega} f(x) \tag{5.1}$$

where $\Omega = \Omega_1 \times \dots \Omega_m$, with $\Omega_i \subseteq \mathbb{R}^{n_i}$ closed and convex subset, $\sum_{i=1}^m n_i = n$, so that any $x \in \Omega$ can be block partitioned as $x = (x_1^T, \dots, x_m^T)^T$, $x_i \in \mathbb{R}^{n_i}$, and $f : \Omega \to \mathbb{R}$ is a continuously differentiable function.

Problem (5.1) is typically tackled by using the *nonlinear Gauss-Seidel* (GS) method [22, p. 267], also known as *nonlinear block coordinate descent* or *alternating optimization* method, which is based on the idea of performing successive minimizations over each block of the function f. In particular, at each iteration $k \in \mathbb{N}$, the iterate $x^{(k+1)} = (x_1^{(k+1)}, \dots, x_m^{(k+1)})$ is computed such that each component $x_i^{(k+1)}$, $i = 1, \dots, m$, is a solution of the subproblem

$$x_i^{(k+1)} \in \operatorname*{argmin}_{x\in\Omega_i} f(x_1^{(k+1)}, \dots, x_{i-1}^{(k+1)}, x, x_{i+1}^{(k)}, \dots, x_m^{(k)}). \tag{5.2}$$

The GS method is often useful in applications where the objective function and the constraints have a partially decomposable structure, such as nonnegative matrix factorization [87, 92] or blind deconvolution [113, 114].

The convergence of the GS scheme has been studied in several works [109, 76, 77, 139, 23] and is guaranteed in the following cases:

- if $m = 2$, then each limit point of the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ generated by the GS method is stationary, without further assumptions nor on the subproblem (5.2) or the objective function [77];

- if $m \geq 3$, one has either to ask that each subproblem (5.2) has a unique solution for $i = 1, \dots, m$, or impose some additional convexity assumptions (i.e. that f is pseudoconvex or f is componentwise strictly quasi convex with respect to $m-2$ blocks of variables) in order to guarantee the stationarity of the limit points [77, 139]; without one of these assumptions holding, the GS method may fail to locate stationary points, as the famous Powell's counterexample shows [109].

Note that the aforementioned requirements on the objective function and the subproblems (5.2) are quite restrictive and, in addition, computing an exact minimum of f, even if restricted to a single block, can be impractical. To overcome these limitations, effective methods capable of handling general nonconvex problems and with global convergence properties have been devised by performing *inexactly* the partial minimization over each block of variables [30, 39, 76].

In this light, we further develop the cyclic block gradient projection method proposed in [30], allowing for generalized projections based on non Euclidean distances. In particular, we propose a block coordinate gradient projection method which, at each outer iteration k, applies a finite number of inner iterations of the form

$$x^{(k+1)} = x^{(k)} + \lambda_k (y^{(k)} - x^{(k)})$$

to each subproblem of type (5.2), where $\lambda_k \in (0,1]$ is the Armijo line–search parameter and $y^{(k)}$ is defined as

$$y^{(k)} = \underset{y \in \mathbb{R}^n}{\operatorname{argmin}}\, h_{\sigma^{(k)}}(y, x^{(k)})$$

where h_σ is a suitable convex function depending on the array of parameters $\sigma \in \mathbb{R}^q$. We show that any limit point of the generated sequence is stationary without any convexity assumption. Our general framework includes, but it is not limited to, several state-of-the-art methods, such as the scaled gradient projection method [37], the spectral projected gradient method [25], the cyclic block gradient projection method [30] and the successive convex approximation algorithm [123].

The outline of the chapter is now detailed. Section 5.1 is concerned with the analysis of the proposed block coordinate descent algorithm: in particular, in Section 5.1.1 we devise the properties of the operator h_σ, which allow to define a class of generalized projection operators, whereas in Section 5.1.2 we present the algorithm and develop the related convergence analysis. Section 5.2 is devoted to some illustrative numerical examples in image blind deconvolution from a single image. Finally, in Section 5.3 we apply our method to the problem of Poisson blind deconvolution from multiple images.

5.1 The proposed algorithm

5.1.1 Generalized gradient projections

In this section we give the definition of a generalized projection operator, providing some examples of well-known functions belonging to this category.

Definition 5.1. *Let $S \subseteq \mathbb{R}^q$. A family of metric functions associated to f on Ω is any set of the form $\mathcal{H}(f, \Omega, S) = \{h_\sigma\}_{\sigma \in S}$ where, for any choice of the parameter $\sigma \in S$, the function $h_\sigma : \Omega \times \Omega \to \mathbb{R}$ satisfies the following properties:*

(H1) h_σ is continuously differentiable;

(H2) h_σ is convex with respect to its first argument, i.e.

$$h_\sigma(y, z) \geq h_\sigma(x, z) + \nabla_1 h_\sigma(x, z)^T (y - x) \quad \forall x, y, z \in \Omega \tag{5.3}$$

where $\nabla_1 h_\sigma(x, z)$ is the gradient of $h_\sigma(\cdot, z)$ at the point x and, for any $z \in \Omega$, $h_\sigma(\cdot, z)$ admits a unique minimum point;

(H3) for any point $x \in \Omega$ and for any feasible direction $d \in \mathbb{R}^n$ we have

$$\nabla_1 h_\sigma(x, x)^T d = \nabla f(x)^T d; \tag{5.4}$$

(H4) h_σ continuously depends on the parameter σ.

Furthermore, the associated generalized gradient projection operator $p(\,\cdot\,; h_\sigma) : \Omega \to \Omega$ *is defined as*

$$p(x; h_\sigma) = \arg\min_{z \in \Omega} h_\sigma(z, x) \quad \forall x \in \Omega. \tag{5.5}$$

Example 5.1. Properties (5.3)–(5.4) are satisfied when the function h_σ is defined as

$$h_\sigma(x, y) = \nabla f(y)^T (x - y) + d_\sigma(x, y), \tag{5.6}$$

where $d_\sigma \in \mathcal{D}(\Omega, S)$ is a distance-like function Ω in the sense of Definition 3.1. In these settings we can find:

a) the standard Euclidean projection $p(x; h_\sigma) = P_\Omega(x - \sigma \nabla f(x))$, obtained by choosing

$$d_\sigma(x, y) = \frac{1}{2\sigma} \|x - y\|^2, \quad \sigma > 0; \tag{5.7}$$

b) the scaled Euclidean projection $p(x; h_\sigma) = P_{\Omega,D}(x - \alpha D^{-1} \nabla f(x))$ corresponding to the choice

$$d_{(\alpha,D)}(x, y) = \frac{1}{2\alpha} (x - y)^T D (x - y). \tag{5.8}$$

In this case the array of parameters σ is given by the pair (α, D), where $\alpha > 0$ and $D \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix;

c) the Bregman distance associated to a strongly convex function $\psi : \Omega \to \mathbb{R}$, which is defined as

$$d_\sigma(x, y) = \frac{1}{\sigma} (\psi(x) - \psi(y) - \nabla \psi(y)^T (x - y)), \quad \sigma > 0. \tag{5.9}$$

Example 5.2. If f is convex, a further class of functions satisfying the properties of Definition 5.1 is given by

$$h_\sigma(x,y) = f(x) + d_\sigma(x,y), \tag{5.10}$$

where again $d_\sigma \in \mathcal{D}(\Omega, S)$. If d_σ is chosen as in (5.7), the resulting $p(\cdot, h_\sigma)$ is the proximal operator of f.

Example 5.3. Let $f = f_0 + f_1$, where $f_0, f_1 : \Omega \to \mathbb{R}$ are both continuously differentiable and f_1 is convex. The metric function given in Definition (3.5), i.e.

$$h_\sigma(x,y) = \nabla f_0(y)^T(x-y) + d_\sigma(x,y) + f_1(x) - f_1(y) \quad \forall x, y \in \Omega, \tag{5.11}$$

with $d_\sigma \in \mathcal{D}(\Omega, S)$, belongs to $\mathcal{H}(f, \Omega, S)$. The associated operator $p(\cdot, h_\sigma)$ is the generalized forward–backward operator defined in Chapter 3. If d_σ reduces to (5.7), one recovers the proximal–gradient operator.

Any function $h_\sigma \in \mathcal{H}(f, \Omega, S)$ can be exploited to define a descent direction for problem (5.1), as stated in the following proposition.

Proposition 5.1. *Let $x \in \Omega$, $\sigma \in S \subseteq \mathbb{R}^q$, $h_\sigma \in \mathcal{H}(f, \Omega, S)$ and*

$$y = p(x; h_\sigma). \tag{5.12}$$

Then we have that

$$\nabla f(x)^T(y - x) \leq 0 \tag{5.13}$$

and the equality holds if and only if $y = x$.

Proof. Inequality (5.3) with $z = x$ yields

$$\nabla_1 h_\sigma(x,x)^T(y-x) \leq h_\sigma(y,x) - h_\sigma(x,x) \leq 0,$$

where the rightmost inequality follows from (5.5) and, since the minimum point of $h_\sigma(\cdot, x)$ is unique, the equality holds if and only if $x = y$. Then, the thesis follows recalling (5.4). □

In the following proposition, we show that the stationary points of (5.1) can be characterized as fixed points of the generalized projection operator (5.5).

Proposition 5.2. *Let $S \subseteq \mathbb{R}^q$, $\sigma \in S$ and $h_\sigma \in \mathcal{H}(f, \Omega, S)$. A point $x \in \Omega$ is a stationary point for problem* (5.1) *if and only if $x = p(x; h_\sigma)$.*

Proof. Assume that for a point $x^* \in \Omega$ the following equality holds:

$$x^* = \arg\min_{x \in \Omega} h_\sigma(x, x^*).$$

Then, the stationarity of x^* yields

$$\nabla_1 h_\sigma(x^*, x^*)^T(x - x^*) \geq 0 \quad \forall x \in \Omega.$$

Since by assumption (5.4) we have $\nabla_1 h_\sigma(x^*, x^*)^T(x - x^*) = \nabla f(x^*)^T(x - x^*)$, it follows that x^* is a stationary point for problem (5.1).
Conversely, let $x^* \in \Omega$ be a stationary point of (5.1) and define

$$\bar{x} = \arg\min_{x \in \Omega} h_\sigma(x, x^*).$$

Assume by contradiction that $x^* \neq \bar{x}$. Then, combining (5.3) with $x = z = x^*$, $y = \bar{x}$ and (5.4) we obtain

$$\nabla f(x^*)^T(\bar{x} - x^*) \leq h_\sigma(\bar{x}, x^*) - h_\sigma(x^*, x^*) < 0,$$

where the last inequality follows from the fact that $\bar{x}$ is the unique minimum point of $h_\sigma(\cdot, x^*)$ and $x^* \neq \bar{x}$. This contradicts the stationarity assumption on x^*. □

5.1.2 Algorithm and convergence analysis

In this section we consider problem (5.1) where the constraint set has the following separable structure

$$\Omega = \Omega_1 \times \ldots \Omega_m, \ \Omega_i \subseteq \mathbb{R}^{n_i}, \ \sum_{i=1}^{m} n_i = n \tag{5.14}$$

so that any $x \in \Omega$ can be block partitioned as $x = (x_1^T, \ldots, x_m^T)^T$, $x_i \in \mathbb{R}^{n_i}$.

Algorithm BLS Block Armijo linesearch algorithm

Let $\{z^{(k)}\}_{k \in \mathbb{N}}$ be a sequence of points in Ω and $\{d_i^{(k)}\}_{k \in \mathbb{N}}$ a sequence of descent directions, for a given $i \in \{1, ..., m\}$. Fix $\delta_i, \beta \in (0, 1)$ and compute $\lambda_i^{(k)}$ as follows:

```
1. Set $\lambda_i^{(k)} = 1$;
2. IF
       $f(z_1^{(k)}, ..., z_i^{(k)} + \lambda_i^{(k)} d_i^{(k)}, ..., z_m^{(k)}) \leq f(z^{(k)}) + \beta \lambda_i^{(k)} \nabla_i f(z^{(k)})^T d_i^{(k)}$     (5.15)
   THEN go to step 3.
   ELSE set $\lambda_i^{(k)} = \delta_i \lambda_i^{(k)}$ and go to step 2.
3. END
```

The key ingredients of our approach are the sufficient decrease of the objective function enforced by a block version of the classical Armijo backtracking procedure (1.12), which is reported in Algorithm BLS, and a suitable metric function $h_\sigma \in \mathcal{H}(f,\Omega,S)$ defined so that it is separable with respect to the partition in (5.14).

In the following proposition we give conditions which guarantee that Algorithm BLS is well defined, and can be considered as a special case of Proposition 3.4, from which its proof can be derived.

Proposition 5.3. *Let $\{z^{(k)}\}_{k\in\mathbb{N}}$ be a sequence of points in Ω. Assume that $z^{(k)}$ converges to some $\bar{z}$ and for $i \in \{1,...,m\}$ let $\{d_i^{(k)}\}_{k\in\mathbb{N}}$ be a sequence of feasible directions such that*

(A1) there exists a number $M > 0$ such that $\|d_i^{(k)}\| \leq M$ for all $k \in \mathbb{N}$;

(A2) we have $\nabla_i f(z^{(k)})^T d_i^{(k)} < 0$ for all $k \in \mathbb{N}$, where $\nabla_i f(z^{(k)})$ denotes the partial gradient of f w.r.t. the $i-$th block of variables at the point $z^{(k)}$;

(A3) we have $\lim\limits_{k\to\infty} f(z^{(k)}) - f(z_1^{(k)},...,z_i^{(k)} + \lambda_i^{(k)} d_i^{(k)},...,z_m^{(k)}) = 0$, where $\lambda_i^{(k)}$ is computed with Algorithm BLS.

Then, for each $k \in \mathbb{N}$ the LS procedure terminates in a finite number of steps and, furthermore, $\lim_{k\to\infty} \nabla_i f(z^{(k)})^T d_i^{(k)} = 0$.

In order to formally introduce the proposed method, we choose the metric function $h_\sigma \in \mathcal{H}(f,\Omega,S)$, where $S = S_1 \times ... \times S_m$, $S_i \subset \mathbb{R}^{q_i}$, such that the parameter σ can be partitioned as $\sigma = (\sigma_1,\ldots,\sigma_m)$. Moreover, we define h_σ so that it is separable over the m blocks with respect to its first variable, i.e.

$$h_\sigma(x,y) = \sum_{i=1}^{m} h_{\sigma_i}^i(x_i,y), \tag{5.16}$$

where the functions $h_{\sigma_i}^i : \Omega_i \times \Omega \to \mathbb{R}$ satisfy the following conditions:

(BH1) $h_{\sigma_i}^i$ is continuously differentiable;

(BH2) $h_{\sigma_i}^i$ is convex with respect to its first argument and admits a unique minimum point;

(BH3) for any point $x \in \Omega$ and for any vector $d \in \mathbb{R}^{n_i}$ such that $x_i + d \in \Omega_i$ we have

$$\nabla_1 h_{\sigma_i}^i(x_i,x)^T d = \nabla_i f(x)^T d, \tag{5.17}$$

where $\nabla_i f(x)$ denotes the gradient of f with respect to the i–th block of variables;

(BH4) $h_{\sigma_i}^i$ continuously depends on the parameter $\sigma_i \in \mathbb{R}^{q_i}$.

It is easy to see that the metric function h_σ defined in (5.16), thanks to the assumptions (BH1)–(BH4), belongs to $\mathcal{H}(f,\Omega,S)$ and the associated generalized gradient projection can be also partitioned by blocks as

$$p(x;h_\sigma) = \begin{pmatrix} p_1(x;h_{\sigma_1}^1) \\ \vdots \\ p_m(x;h_{\sigma_m}^m) \end{pmatrix}, \quad \text{where} \quad p_i(x;h_{\sigma_i}^i) = \arg\min_{z_i\in\Omega_i} h_{\sigma_i}^i(z_i,x). \tag{5.18}$$

Lemma 5.1. *Let $x \in \Omega$ and $\sigma \in S \subseteq \mathbb{R}^q$. Then,*

(i) x is stationary for problem (5.1) *if and only if $p_i(x; h^i_{\sigma_i}) = x_i \ \forall i = 1, \dots, m$;*

(ii) $\nabla_i f(x)^T (p_i(x; h^i_{\sigma_i}) - x_i) \leq 0 \ \forall i = 1, \dots, m$ and the equality holds if and only if $x_i = p_i(x; h^i_{\sigma_i})$.

Proof. Part (i) of the previous Lemma directly follows from (5.18) and from Proposition 5.2, while part (ii) can be easily proved by employing the same arguments as in the proof of Proposition 5.1. □

Algorithm CBGGP Cyclic Block Generalized Gradient Projection Method

```
Define a compact set $S$ and a metric $h_\sigma \in \mathcal{H}(f, \Omega, S)$ as in (5.16). Choose $\beta, \delta \in (0,1)$.
Choose $x^{(0)} \in \Omega$ and the upper bounds for the inner iterations numbers $L_1, \dots, L_m$.
FOR $k = 0, 1, 2, ...$
    1 Set $z(k,0) = x^{(k)}$
    2 FOR $i = 1, ..., m$
        2.1 Set $x_i^{(k,0)} = x_i^{(k)}$
        2.2 Choose the inner iterations number $L_i^{(k)} \leq L_i$
        2.3 FOR $\ell = 0, ..., L_i^{(k)} - 1$
          2.3.0 Set $\tilde{x}^{(k,\ell)} = (x_1^{(k+1)}, \dots, x_{i-1}^{(k+1)}, x_i^{(k,\ell)}, x_{i+1}^{(k)}, \dots, x_m^{(k)})$
          2.3.1 Choose the parameter $\sigma_i^{(k,\ell)} \in S_i$
          2.3.2 Compute the descent direction $d_i^{(k,\ell)} = p_i(\tilde{x}^{(k,\ell)}; h^i_{\sigma_i^{(k,\ell)}}) - x_i^{(k,\ell)}$ and set $\tilde{d}^{(k,\ell)} =$
                $(0, \dots, 0, d_i^{(k,\ell)}, 0, \dots, 0)$
          2.3.3 Compute with Algorithm BLS the Armijo steplength $\lambda_i^{(k,\ell)}$ such that
                $f(\tilde{x}^{(k,\ell)} + \lambda_i^{(k,\ell)} \tilde{d}^{(k,\ell)}) \leq f(\tilde{x}^{(k,\ell)}) + \beta \lambda_i^{(k,\ell)} \nabla_i f(\tilde{x}^{(k,\ell)})^T d_i^{(k,\ell)}$
          2.3.4 Set $x_i^{(k,\ell+1)} = x_i^{(k,\ell)} + \lambda_i^{(k,\ell)} d_i^{(k,\ell)}$
            END
        2.4 Set $x_i^{(k+1)} = x_i^{(k, L_i^{(k)})}$
        2.5 Set $z(k,i) = (x_1^{(k+1)}, ..., x_i^{(k+1)}, x_{i+1}^{(k)}, ..., x_m^{(k)})$
      END
    3 Set $x^{(k+1)} = z(k,m)$
END
```

The previous results can be exploited to design a cyclic block generalized gradient projection (CBGGP) method [36, 126], whose steps are outlined in Algorithm CBGGP. Before analysing the convergence properties of this approach, we observe that it is a descent method and, in particular, the objective function is nonincreasing over the *partial updates* $z(k,i)$, $i = 0, ..., m$, $k = 1, 2, ...$ defined at step 2.5. Indeed, the following inequalities hold

$$f(z(k,i+1)) \leq f(z(k,i)) + \beta\lambda_{i+1}^{(k,0)} \nabla_{i+1} f(z(k,i))^T d_{i+1}^{(k,0)} \leq f(z(k,i))$$

which also implies

$$\begin{aligned} f(z(k+1,0)) = f(z(k,m)) &\leq f(z(k,i+1)) \\ &\leq f(z(k,i)) \leq f(z(k,0)) = f(z(k-1,m)). \end{aligned} \tag{5.19}$$

We are now ready to give the first result about Algorithm CBGGP.

Proposition 5.4. *Let $\{x^{(k)}\}_{k\in\mathbb{N}}$ be the sequence generated by Algorithm CBGGP. Suppose that for some $i \in \{0, ..., m\}$ the sequence $\{z(k,i)\}_{k\in\mathbb{N}}$ admits a limit point $\bar{z}$. Then $p_{i+1}(\bar{z}; h_{\sigma_{i+1}}^{i+1}) = \bar{z}_{i+1}$ $\forall \sigma_{i+1} \in S_{i+1}$ if $i < m$, while $p_1(\bar{z}; h_{\sigma_1}^1) = \bar{z}_1$ $\forall \sigma_1 \in S_1$ if $i = m$.*

Proof. Suppose first that $i < m$. From Lemma 5.1, we only need to show that there exists $\bar{\sigma}_{i+1} \in S_{i+1}$ such that equality $p_{i+1}(\bar{z}; h_{\bar{\sigma}_{i+1}}^{i+1}) = \bar{z}_{i+1}$ holds.
Assume by contradiction that $p_{i+1}(\bar{z}; h_{\sigma_{i+1}}^{i+1}) \neq \bar{z}_{i+1}$ for all $\sigma_{i+1} \in S_{i+1}$. Let K be the set of indices such that $\{z(k,i)\}_{k\in K}$ converges to $\bar{z}$ and $\{\sigma_{i+1}^{(k,0)}\}_{k\in K}$ converges to some $\bar{\sigma}_{i+1} \in S_{i+1}$. If $\|p_{i+1}(\bar{z}; h_{\sigma_{i+1}}^{i+1}) - \bar{z}_{i+1}\| = 2\epsilon > 0$, the continuity of the generalized projection operator with respect to all its arguments guarantees that, for $k \in K$ being sufficiently large, we have

$$\|d_{i+1}^{(k,0)}\| > \epsilon > 0,$$

where $d_{i+1}^{(k,0)} = p_{i+1}(z(k,i); h_{\sigma_{i+1}^{(k,0)}}^{i+1}) - x_{i+1}^{(k)}$ (see also Step 2.3.2 of Algorithm CBGGP). Then, by applying Lemma 5.1 (ii) we have

$$\nabla_{i+1} f(z(k,i))^T d_{i+1}^{(k,0)} \leq -\eta < 0, \tag{5.20}$$

where η is some positive scalar. On the other hand, inequalities (5.19) guarantee that, for all i, we have $\lim_{k\to\infty} f(z(k,i)) = f(\bar{z})$, thus we obtain that

$$\lim_{k\to\infty} f(z(k,i)) - f(x_1^{(k+1)}, ..., x_i^{(k+1)}, x_{i+1}^{(k)} + \lambda_{i+1}^{(k,0)} d_{i+1}^{(k,0)}, ..., x_m^{(k)}) = 0.$$

Moreover, since $\{z(k,i)\}_{k\in K}$ is a convergent sequence, it is also bounded. Therefore the sequence $\{d_{i+1}^{(k,0)}\}_{k\in K}$ is bounded and Proposition 5.3 implies that

$$\lim_{k\to\infty, k\in K} \nabla_{i+1} f(z(k,i))^T d_{i+1}^{(k,0)} = 0,$$

which contradicts (5.20).
The same arguments can be applied also when $i = m$, since $z(k,m) = z(k+1,0)$. □

The previous proposition is crucial for proving the main convergence result for Algorithm CBGGP, given below.

Theorem 5.1. *Let* $\{x^{(k)}\}_{k\in\mathbb{N}}$ *be the sequence generated by Algorithm CBGGP and assume that* $\bar{x}$ *is a limit point of* $\{x^{(k)}\}_{k\in\mathbb{N}}$*. Then* $\bar{x}$ *is a limit point also for the sequences* $\{z(k,i)\}_{k\in\mathbb{N}}$ *for any* $i = 1, ..., m-1$ *and it is a stationary point for problem* (5.1)*.*

Proof. The proof runs by induction on the block index i and on the inner iteration number ℓ and it is similar to that of Theorem 4.2 in [30]. Since $\bar{x}$ is a limit point for $\{x^{(k)}\}_{k\in\mathbb{N}} = \{z(k,0)\}_{k\in\mathbb{N}}$, from Proposition 5.4 it follows that, denoting by K_0 a set of indices such that $\{x^{(k)}\}_{k\in K_0}$ converges to $\bar{x}$ and $\{\sigma_1^{(k,0)}\}_{k\in K_0}$ converges to some $\bar{\sigma}_1^0 \in S_1$, we have $p_1(\bar{x}; h^1_{\bar{\sigma}_1^0}) = \bar{x}_1$ and $\lim_{k\to\infty, k\in K_0} \|d_1^{(k,0)}\| = 0$.
From step 2.3.4 of Algorithm CBGGP, it follows that $\lim_{k\to\infty, k\in K_0} \|x_1^{(k,1)} - x_1^{(k)}\| = 0$, i.e., $\bar{x}_1$ is a limit point also for the sequence $\{x_1^{(k,1)}\}_{k\in\mathbb{N}}$.
Introducing a subset of indices $K_1 \subseteq K_0$ such that the sequence $\{x_1^{(k,1)}\}_{k\in K_1}$ converges to $\bar{x}_1$ and $\{\sigma_1^{(k,1)}\}_{k\in K_1}$ converges to some $\bar{\sigma}_1^1$, we have

$$\begin{aligned}\lim_{k\to\infty, k\in K_1} d_1^{(k,1)} &= \lim_{k\to\infty, k\in K_1} p_1((x_1^{(k,1)}, x_2^{(k)}, ..., x_m^{(k)}); h^1_{\sigma_1^{(k,1)}}) - x_1^{(k,1)} \\ &= p_1(\bar{x}; h^1_{\bar{\sigma}_1^1}) - \bar{x}_1 = 0,\end{aligned}$$

where the second equality follows from the continuity of the generalized projection operator and the third one is a consequence of Proposition 5.4.
Using the same arguments, by induction on ℓ we can conclude that, for each $\ell = 0, ..., L_1 - 1$, there exists a suitable subset of indices K_ℓ such that $\lim_{k\to\infty, k\in K_\ell} d_1^{(k,\ell)} = 0$ and we obtain

$$\|x_1^{(k+1)} - x_1^{(k)}\| \leq \sum_{\ell=0}^{L_1^{(k)}} \lambda_1^{(k,\ell)} \|d_1^{(k,\ell)}\| \leq \sum_{\ell=0}^{L_1} \lambda_1^{(k,\ell)} \|d_1^{(k,\ell)}\| \xrightarrow{k\to\infty, k\in\bar{K}_1} 0,$$

where $\bar{K}_1 = \cap_{\ell=0}^{L_1-1} K_\ell$. Thus, the point $\bar{x}$ is a limit point also for the sequence $\{z(k,1)\}_{k\in\mathbb{N}} = \{(x_1^{(k+1)}, x_2^{(k)}, ..., x_m^{(k)})\}_{k\in\mathbb{N}}$, and Proposition 5.4 ensures that $p_2(\bar{x}; h^2_{\bar{\sigma}_2^0}) = \bar{x}_2$ for some $\bar{\sigma}_2^0 \in S_2$. Proceeding by induction on i and employing the same arguments used for $i = 1$, we prove that $\bar{x}$ is a limit point of the sequences $\{z(k,i)\}_{k\in\mathbb{N}}$ for any $i = 1, ..., m-1$. As a result of this, invoking again Proposition 5.4, we can conclude that for any $i = 1, ..., m$ there exist $\sigma_i \in S_i$ such that $p_i(\bar{x}; h^i_{\sigma_i}) = \bar{x}_i$. Therefore, by Lemma 5.1 (i) we can conclude that $\bar{x}$ is a stationary point of problem (5.1). □

5.2 Application in image blind deconvolution from a single image

In this section we consider a relevant application and we show that it can be effectively solved by algorithms which can be framed in the analysis of the previous sections. We give also some guidelines on how to choose the parameter σ at each iteration. The application we consider is the image blind deconvolution problem in presence of either Gaussian or Poisson noise. Our basic assumption is that the available data $g \in \mathbb{R}^{p^2}$ is a realization of a Gaussian/Poisson random variable whose mean is $\overline{\omega} \otimes \overline{f} + b\boldsymbol{e}$, where $\overline{\omega} \in \mathbb{R}^{p^2}$ is an unknown point spread function (PSF), $\otimes$ denotes the convolution operator (periodic boundary conditions are assumed), b is a positive parameter representing the background radiation, $\boldsymbol{e} \in \mathbb{R}^{p^2}$ is the vector of all ones and $\overline{f} \in \mathbb{R}^{p^2}$ is the image we would like to recover. In the following, we will assume that the PSF $\overline{\omega}$ is normalized to one.

5.2.1 Gaussian noise

For the Gaussian noise tests we follow a maximum a posteriori approach [20] and consider the optimization problem

$$\min_{f \in \Omega_f, \omega \in \Omega_\omega} J(f, \omega) \equiv LS(f, \omega) + \rho_1 R_1(f) + \rho_2 R_2(\omega), \tag{5.21}$$

where LS is the least-squares distance

$$LS(f, \omega) = \frac{1}{2} \|\omega \otimes f + b\boldsymbol{e} - g\|_2^2, \tag{5.22}$$

ρ_1, ρ_2 are positive regularization parameters and R_1, R_2 are differentiable and convex regularization terms. The feasible sets Ω_f and Ω_ω have been chosen according to the physical features of the imaging problem, since we restricted the analysis to non-negative images f and non-negative and normalized PSFs:

$$\begin{aligned} \Omega_f &= \{f \in \mathbb{R}^{p^2} \mid f \geq 0\}, \\ \Omega_\omega &= \{\omega \in \mathbb{R}^{p^2} \mid \omega \geq 0, \sum_{i=1}^{p^2} \omega_i = 1\}. \end{aligned}$$

We considered two images, called “satellite” and “crab”, the former one being the satellite image frequently used in several papers on image deblurring [5, 38, 137] and the latter one being the Hubble Space Telescope (HST) image of the crab nebula NGC 1952, exploited in astronomical image deconvolution tests [18, 19, 52, 112]. The 256 × 256 satellite image has values in the range $[0, 2.52 \cdot 10^{-4}]$ and has been artificially blurred with an out-of-focus PSF with radius equal to 4. A constant background equal to one tenth of $\max(\overline{f})$ has been added to the resulting image before corrupting it with 5% Gaussian noise. For this dataset, we chose the two regularization terms R_1 and R_2 equal to the hypersurface potential [2, 19]

$$R_{HS}(x) = \sum_{i,j=1}^{p} \sqrt{((\mathcal{D}x)_{i,j})_1^2 + ((\mathcal{D}x)_{i,j})_2^2 + \nu^2}, \tag{5.23}$$

where ν is a positive parameter and the discrete gradient operator $\mathcal{D} : \mathbb{R}^{p^2} \longrightarrow \mathbb{R}^{2p^2}$ is set through the standard finite difference with periodic boundary conditions

$$(\mathcal{D}x)_{i,j} = \begin{pmatrix} ((\mathcal{D}x)_{i,j})_1 \\ ((\mathcal{D}x)_{i,j})_2 \end{pmatrix} = \begin{pmatrix} x_{i+1,j} - x_{i,j} \\ x_{i,j+1} - x_{i,j} \end{pmatrix}, \quad x_{p+1,j} = x_{1,j}, \quad x_{i,p+1} = x_{i,1}. \tag{5.24}$$

As concerns the crab dataset, the target image is still sized 256 × 256 and has values in the range $[0, 3.58 \cdot 10^5]$. The procedure used to produce the noisy crab image is that exploited for the satellite, but in this case the original image has been blurred with an Airy function [1] mimicking the ideal acquisition of one mirror of the Large Binocular Telescope (LBT – http://www.lbto.org). Due to the smoother content of the image to be restored, for the crab dataset we used as regularization terms R_1 and R_2 the Tikhonov functional [138]

$$R_T(x) = \|x\|_2^2. \tag{5.25}$$

The original and noisy images for both the satellite and the crab datasets are shown in Figure 5.1.

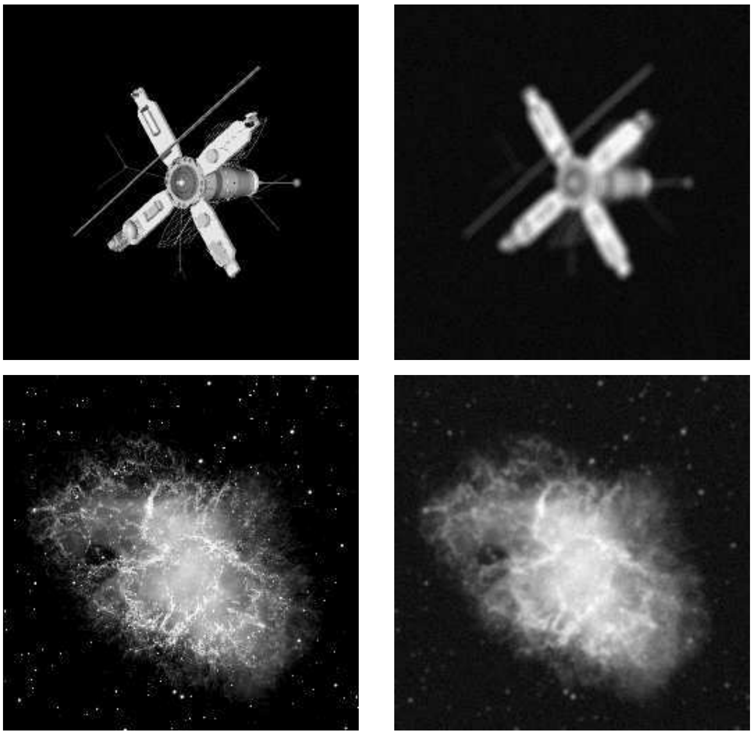


Figure 5.1: Satellite (top) and crab (bottom) test problems: original target (left) and blurred and noisy image (right).

As for the reconstruction algorithms, thanks to the separability of the constraints on f and ω, we used the approach described in Algorithm CBGGP. We remark that, at each iteration, one has to sequentially address two minimization subproblems of the kind

$$\min_{u \in \Omega} J(u) \tag{5.26}$$

where $u \in \mathbb{R}^{p^2}$ represents, alternately, the astronomical object f or the PSF ω, Ω indicates either Ω_f or Ω_ω and $J(u)$ is the objective function of (5.21) with respect to one of the two unknown block of variables only, keeping the other one fixed. Each of these subproblems will solved by a fixed number of iterations of the form

$$u^{(k+1)} = u^{(k)} + \lambda_k (p(u^{(k)}; h_{\sigma^{(k)}}) - u^{(k)})$$

where h_σ is one of the three metric functions detailed in Example 5.1, λ_k is the Armijo parameter and $\sigma^{(k)}$ is a steplength parameter adaptively computed (here k denotes the index of the inner iterations).
The main features of the three gradient projection operators are detailed below.

Scaled gradient projection (SGP): in this case we pick $p(u^{(k)}; h_{\sigma^{(k)}}) = P_{\Omega, D_k}(u^{(k)} - \alpha_k D_k^{-1} \nabla f(u^{(k)}))$, where $D_k \in \mathbb{R}^{p^2 \times p^2}$ is a symmetric positive definite matrix and $\alpha_k > 0$. As concerns the scaling matrix D_k, we exploit the gradient decomposition technique (see Section 1.2.2 and 3.3) applied to the least squares term in (5.21). The resulting diagonal scaling matrix is

$$[D_k]_{ii}^{-1} = \max \left\{ \frac{1}{\mu}, \min \left\{ \mu, \frac{u_i^{(k)}}{\left(A^T(Au^{(k)} + b\boldsymbol{e})\right)_i} \right\} \right\},$$

where μ is a prefixed threshold, and A is the block circulant with circulant blocks matrix computed on the fixed unknown (i.e., in the minimization step over the image f, we have $Af = \omega^{(k)} \otimes f$, while in the minimization step over the PSF ω we have $A\omega = \omega \otimes f^{(k)}$). The steplength parameter α_k is then computed by the adaptive alternation of the scaled Barzilai–Borwein (BB) rules as proposed in Algorithm 3 (see Chapter 1 and [37]):

$$\alpha_k^{BB1S} = \frac{{s^{(k-1)}}^T D_k D_k s^{(k-1)}}{{s^{(k-1)}}^T D_k y^{(k-1)}} \quad ; \quad \alpha_k^{BB2S} = \frac{{s^{(k-1)}}^T D_k^{-1} y^{(k-1)}}{{y^{(k-1)}}^T D_k^{-1} D_k^{-1} y^{(k-1)}}. \tag{5.27}$$

where $s^{(k-1)} = u^{(k)} - u^{(k-1)}$ and $z^{(k-1)} = \nabla J(u^{(k)}) - \nabla J(u^{(k-1)})$. Finally, the chosen steplength α_k is constrained in an interval $[\alpha_{\min}, \alpha_{\max}]$ with $0 < \alpha_{\min} \leq \alpha_{\max}$.

Gradient projection (GP): this corresponds to set $D_k = I_n$ in the previous settings. The parameter α_k is again computed by alternating the scaled BB rules.

Gradient Bregman projection (GBP): a further instance of projections belonging to the family in Example 5.1 corresponds to the choice of the metric (5.6) with related distance-like function d_σ defined in (5.9), where

$$\psi(u) = \sum_{i=1}^{p^2} (u_i + \gamma) \log(u_i + \gamma),$$

with $\gamma > 0$ is the "regularized entropy" [101]. The resulting projection operator (5.5) is given by

$$[p(u, h_\sigma)]_i = \max\left\{ (u_i + \gamma) e^{-\sigma \nabla_i J(u)} - \gamma, 0 \right\}.$$

The steplength parameter σ is adaptively computed at each iteration in the following way. First, we observe that, by the Taylor expansion of the exponential function, we have

$$(u_i + \gamma) e^{-\sigma \nabla_i J(u)} = (u_i + \gamma) - q_i(\sigma)(u_i + \gamma) \nabla_i J(u),$$

where $q_i(\sigma) = \sum_{j=0}^{\infty} (-1)^j \frac{\sigma^{j+1}}{(j+1)!} \nabla_i J(u)^j$. The term $q_i(\sigma)$ can be explicitly expressed also as $q_i(\sigma) = (1 - e^{-\sigma \nabla_i J(u)}) / \nabla_i J(u)$ when $\nabla_i J(u) \neq 0$, $q_i(\sigma) = \sigma$ when $\nabla_i J(u) = 0$. Then, the GBP method can be considered also an approximated scaled gradient method employing the following scaling matrix

$$[D_k]_{ii}(\sigma) = (u_i^{(k)} + \gamma) q_i(\sigma).$$

Thus, it is reasonable to determine the steplength parameter according to the quasi-Newton approach

$$\min_{\sigma \in [\sigma_{\min}, \sigma_{\max}]} \|D_k(\sigma)^{-1} s^{(k)} - w^{(k)}\|^2, \quad \min_{\sigma \in [\sigma_{\min}, \sigma_{\max}]} \|s^{(k)} - D_k(\sigma) w^{(k)}\|^2,$$

where $s^{(k)} = u^{(k)} - u^{(k-1)}$ and $w^{(k)} = \nabla J(u^{(k)}) - \nabla J(u^{(k-1)})$. The previous one-dimensional minimum problems can be easily solved (for example by means of the `fminbnd` Matlab function), giving two possible values for the steplength σ_k. These values are then alternated by means of the adaptive strategy used for the two Barzilai-Borwein rules in [37] and in the previous two projections.

The initialization $f^{(0)}$ for the image has been set equal to the measured image g for both datasets, while for the PSF we used as $\omega^{(0)}$ an out-of-focus PSF with radius equal to 5.5 for the satellite and a Lorentzian function with half-width at half-maximum equal to 6 for the crab. The regularization parameters (ρ_1, ρ_2) have been arbitrarily fixed equal to $(10^{-3}, 10^{-6})$ for the satellite and $(2 \cdot 10^{-2}, 10^{-6})$ for the crab. The strategy we adopted to arrest the inner iterations in the two minimization subproblems of type (5.26) is that proposed in [30] for the cyclic block gradient projection method, which has been already applied with good outcomes in Gaussian blind deconvolution [52, Section 3.1], as well as in the Poisson case [110]. This rule is based on

Proposition 5.2 and Theorem 5.1, which tell us that any limit point of the CBGGP sequence is a point in which the *generalized projected gradient*

$$\begin{aligned}\nabla^P J(f,\omega) &= (p_f((f,\omega);h_\sigma) - f, p_\omega((f,\omega);h_\sigma) - \omega)\\ &= \left(\nabla_f^P J(f,\omega), \nabla_\omega^P J(f,\omega)\right)\end{aligned}$$

is equal to zero. It follows that a possible stopping rule for the inner iterations applied, for instance, to the subproblem w.r.t. to the block f, can be designed by choosing the first inner iteration ℓ for which $f^{(k-1,\ell)}$ satisfies

$$\|\nabla_f^P J(f^{(k-1,\ell)}, \omega^{(k-1)})\| \le \eta_f^{(k)}$$

where the sequence $\{\eta_f^{(k)}\}_{k\in\mathbb{N}}$ is initialized as $\eta_f^{(0)} = \epsilon \cdot \|\nabla_f^P J(f^{(0)}, \omega^{(0)})\|$ where ϵ is a tolerance parameter, and defined by

$$\eta_f^{(k)} = \begin{cases} 0.1 \cdot \eta_f^{(k-1)}, & \text{if } \eta_f^{(k-1)} \ge \|\nabla_f^P J(f^{(k-1,1)}, \omega^{(k-1)})\|,\\ \eta_f^{(k-1)}, & \text{otherwise.}\end{cases}$$

An analogous rule can be defined for the subproblem referred to the block ω. In few words, the rationale behind the choice of the adaptive parameters $\eta_f^{(k)}$ and $\eta_\omega^{(k)}$ is to decrease the tolerance for the stopping criterion if satisfied at the first inner iteration, thus forcing the inner algorithm to perform at least two steps in each subproblem.

Finally, the outer iterations have been arrested when the relative difference between two successive values of the objective function decreases below 10^{-6} (a maximum number of 200 outer iterations is also imposed).

In Figure 5.2 we show the plots of the relative root mean square errors (RMSEs) $\|f^{(k)} - \overline{f}\|_2/\|\overline{f}\|_2$ and $\|\omega^{(k)} - \overline{\omega}\|_2/\|\overline{\omega}\|_2$ for the three approaches as functions of the outer iterations k, and we can observe that the choice of the projection strongly affect the behaviour of the minimization algorithm, with better performances remarked when a non Euclidean projection is considered. We have to remark that, while the GP and SGP algorithms are computationally almost equivalent (few additional scalar products are needed if a nontrivial scaling matrix D_k is included), the GBP iteration is heavier due to the call of the `fminbnd` Matlab function in the computation of the steplength parameter.

5.2.2 Poisson noise

Although several Poisson blind deconvolution problems have been addressed by a maximum a posteriori approach as done in the previous section (see e.g. [79, 91]), we follow the approach in [113, 114] and consider the nonregularized optimization problem

$$\min_{f\in\Omega_f, \omega\in\Omega_\omega} KL(f,\omega), \tag{5.28}$$

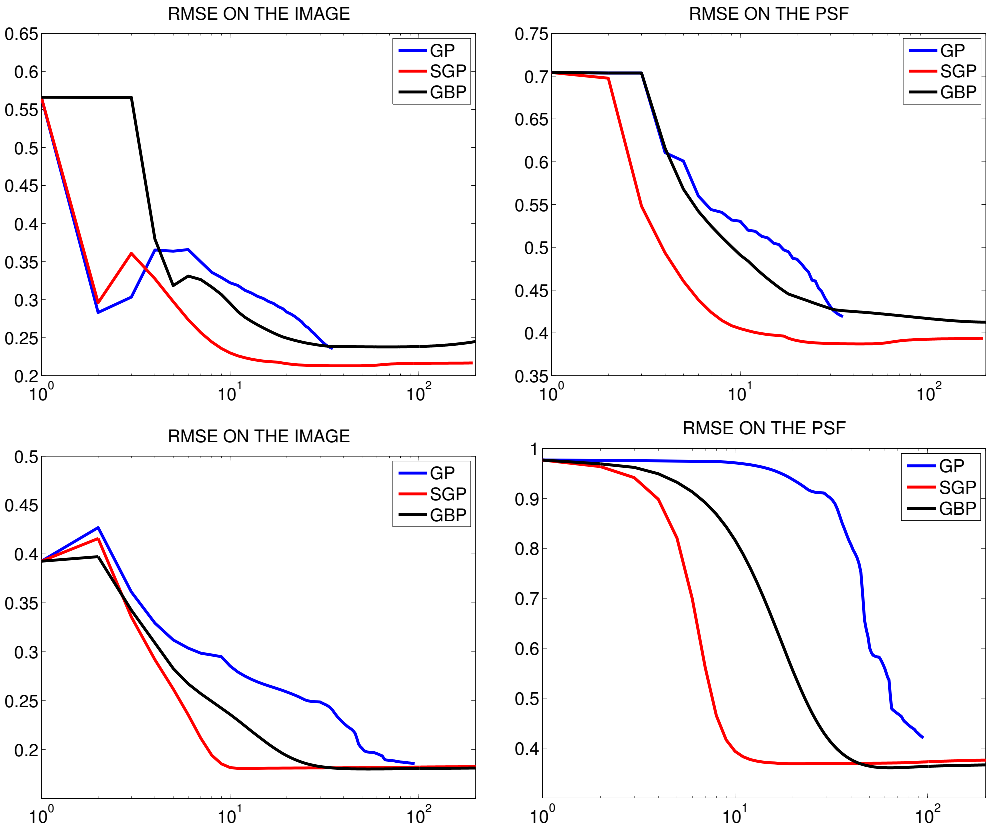


Figure 5.2: Blind deconvolution with the satellite (top) and crab (bottom) test problems: RMSE on image (left) and PSF (right) versus the iterations number.

where KL is the generalized Kullback–Leibler divergence

$$KL(f,\omega) = \sum_{i=1}^{p^2} \left\{ g_i \log\left(\frac{g_i}{(\omega\otimes f)_i + b}\right) + (\omega\otimes f)_i + b - g_i \right\}. \tag{5.29}$$

The choice of avoiding regularization terms is motivated by the fact that, in the following experiments, we will only consider the case of stellar fields or, in other words, of sparse objects, and it is known that the minimizers of the KL divergence already satisfy a sparsity property [18]. In these settings, a regularized solution can be achieved by solving the optimization problem (5.28) approximately through the early stopping of an iterative procedure.

As concerns the feasible sets, we consider non-negativity and flux conservation for the image f, the latter added to further enforce the sparsity of the object, while for the PSF ω we impose non-negativity, normalization to 1 and an upper bound s which can be estimated in the case of adaptive optics devices from the knowledge of the so-called Strehl ratio (SR), i.e. the ratio of peak diffraction intensity of an aberrated versus perfect waveform (see e.g. [97]). The resulting

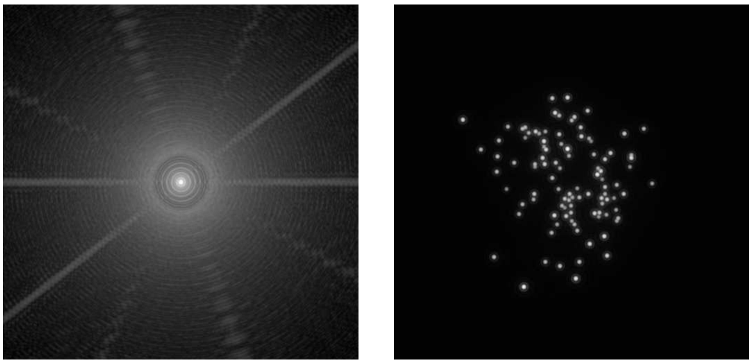


Figure 5.3: Star cluster test problem: original PSF (left) and blurred and noisy image (right). Both images are in log scale.

sets are then given by

$$\Omega_f = \{f \in \mathbb{R}^{p^2} \mid f \geq 0,\ \sum_{i=1}^{p^2} f_i = \sum_{i=1}^{p^2} g_i - p^2 b\},$$

$$\Omega_\omega = \{\omega \in \mathbb{R}^{p^2} \mid 0 \leq \omega \leq s,\ \sum_{i=1}^{p^2} \omega_i = 1\}.$$

These constraints in a blind deconvolution framework have been used e.g. in [59, 113, 114] and allow good reconstructions even in presence of a large scale and nonconvex problem as (5.28).

We consider a realistic simulation in the astronomical field by a) generating a 512 × 512 image $\overline{f}$ of a cluster of 100 stars with different magnitudes (brightest value $\approx 3.2 \cdot 10^7$, dimmest value $\approx 4.2 \cdot 10^6$); b) convolving it with a PSF $\overline{\omega}$ (SR = 0.81) mimicking the response of a single mirror of the large binocular telescope (LBT) and its first light adaptive optics (FLAO) system [60]; c) adding a realistic background radiation ($b \approx 2.6 \cdot 10^4$); and d) corrupting the resulting blurred image with Poisson noise. In Figure 5.3 we reported both the PSF used in this experiment and the simulated measured image.

As done for the Gaussian case, we analyzed the performances of the alternating scheme defined in Algorithm CBGGP with the same three choices for the projection operator described in the previous section. The only change is in the SGP case, for which the scaling matrix used for this test is the one borrowed from the Lucy-Richardson method [95, 128]

$$[D_k]_{ii}^{-1} = \max\left\{\frac{1}{\mu}, \min\left\{\mu, u_i^{(k)}\right\}\right\}, \tag{5.30}$$

suitably thresholded to ensure convergence.

Following the suggestion in [113], we used $L_1^{(k)} = L_1 = 50$ $(k = 1, 2, \ldots)$ inner iterations for the image step, $L_2^{(k)} = L_2 = 1$ $(k = 1, 2, \ldots)$ iteration for the PSF step, a constant image as $f^{(0)}$ and the autocorrelation of the ideal PSF of LBT as $\omega^{(0)}$. The outer iterations have been arbitrarily stopped at 3000.

In Figure 5.4 we show the reconstruction of the PSF provided by the three approaches together with the horizontal and vertical central cuts of the pictures compared with those of the target PSF. Moreover, in Figure 5.5 we plotted the reconstruction errors and the decrease of the objective function versus the number of iterations, where for the PSF we used the standard RMSE $\|\omega^{(k)} - \overline{\omega}\|_2 / \|\overline{\omega}\|_2$ while for the image we computed the RMSE for each star $|f_i^{(k)} - \overline{f}_i| / |\overline{f}_i|$ $(i = 1, \ldots, 100)$ and then calculated the mean of the 100 resulting values. The results obtained in this test problem lead to conclusions quite similar to those drawn up in the previous section, since again different choices for the projection lead to different reconstructions. The SGP and GBP choices seems to be attracted by the same limit point, even if going through different paths. With these approaches the reconstructions are very satisfactory, since both the image and the PSF are restored with an error below 1%. On the contrary, the standard projection in Euclidean norm lead to a significantly different pair (f, ω), with a higher precision in recovering the correct magnitude of the stars coupled with a worse reconstruction of the PSF (RMSE > 20%), as clearly attested also by the plots shown in the second row of Figure 5.4.

5.3 Application in Poisson blind deconvolution from multiple images

We now make a step further from the previous section and assume that the unknown object and PSF(s) must be recovered from a set of multiple images acquired at different rotations of the detection system. In particular, we show numerical experience in which images of binary systems and starts clusters are simulated by adopting the PSF model of the Large Binocular Telescope (LBT).

5.3.1 Problem formulation

Let us assume that $K \geq 1$ different images $\{g_j\}_{j=1}^K$ of the scientific object, with K corresponding expected values of the background emission $\{b_j\}_{j=1}^K$ and K different PSFs $\{\omega_j\}_{j=1}^K$, are available in the Poisson noise model. Since it is quite natural to assume that the K images are statistically independent, the likelihood of the problem is the product of the likelihoods of the different images (see Appendix A). Then, by taking the negative logarithm of the likelihood we obtain the following data-fidelity function which is the sum of K Kullback-Leibler generalized divergences, one for each image, i.e.

$$J_0(f, \omega_1, ..., \omega_K; g, b) = \sum_{j=1}^{K} \sum_{i=1}^{p^2} \left\{ g_j(i) \ln \frac{g_j(i)}{(w_j \otimes f)(i) + b_j(i)} + (\omega_j \otimes f)(i) + b_j(i) - g_j(i) \right\} , \quad (5.31)$$

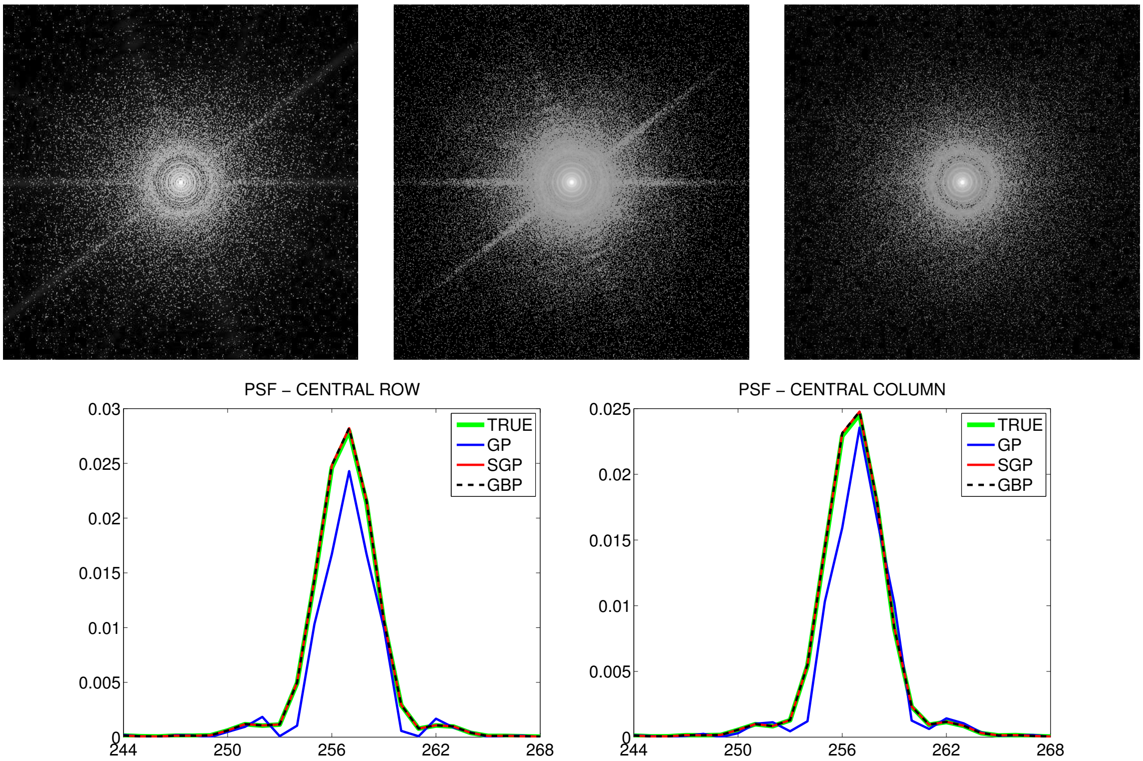


Figure 5.4: PSF for the star cluster test problem. First row: reconstructions with GP (left), SGP (middle) and GBP (right) in log scale. Second row: horizontal and vertical central cuts of original and restored PSFs.

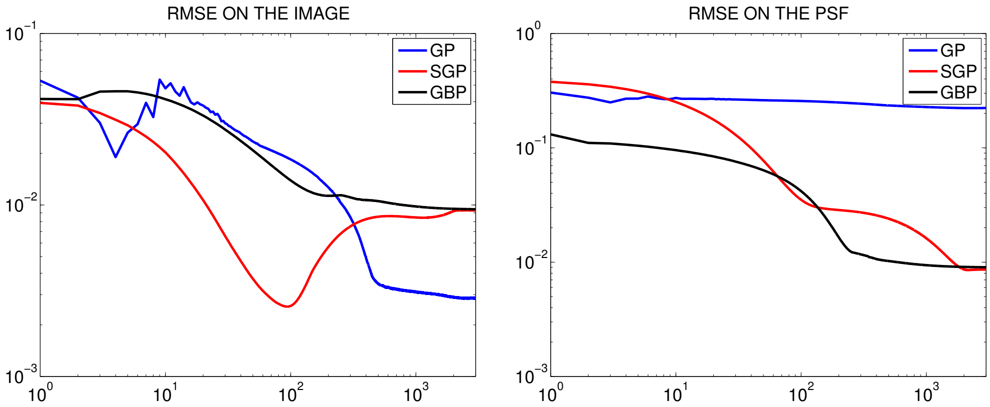


Figure 5.5: Blind deconvolution with the star cluster test problem: RMSE on image (left) and RMSE on PSF (right) in log scale versus the iterations number.

where $(g, b) = \{(g_j, b_j)\}_{j=1}^K$ and the notation $g_j(i)$ denotes the $i-$th component of g_j (the same holds for the other vectors involved). The problem of blind deconvolution consists then in the minimization of (5.31) with respect to $K+1$ blocks of unknown variables, namely the object f and the K PSFs $\{\omega_j\}_{j=1}^K$. The function (5.31) is convex with respect to each block of variables for fixed values of the others, but is not convex with respect to the full set of variables. On our side we have the CBGGP algorithm which, in the light of Theorem 5.1, globally converges also in the nonconvex case, and thus is perfectly suited for this problem.

As for the single image case, some constraints on the unknown object and PSFs must be imposed. As far as the object is concerned, besides non-negativity, we also introduce a constraint on its flux; more precisely we require that the object flux coincides with the average flux of the p detected images (after background subtraction), which is given by

$$c = \frac{1}{K} \sum_{j=1}^{K} \sum_{i=1}^{p^2} \{g_j(i) - b_j(i)\} \quad . \tag{5.32}$$

We remark that this constraint is further enforcing sparsity; in the case of deconvolution and zero value of the backgrounds, it is automatically satisfied by the minimizers of the KL divergence. As concerns the PSFs, as also explained in the previous section, an important constraint is the upper bound derived from the knowledge of the Strehl ratio s_j characterizing the AO correction of the atmospheric blur during the observation. Moreover, nonnegativity and normalization provide additional constraints. In conclusion, the nonconvex optimization problem we are considering can be formulated as follows

$$\min_{f \in \Omega_f, \omega_j \in \Omega_{\omega_j}} J_0(f, \omega_1, \ldots, \omega_K; g, b) \tag{5.33}$$

where

$$\begin{aligned} \Omega_f &= \{f \in \mathbb{R}^{p^2} \mid f \geq 0, \ \sum_{i=1}^{p^2} f_i = c\}, \\ \Omega_{\omega_j} &= \{\omega \in \mathbb{R}^{p^2} \mid 0 \leq \omega_j \leq s_j, \ \sum_{i=1}^{p^2} \omega_j(i) = 1\}, \quad j = 1, \ldots, K. \end{aligned}$$

We address this problem by means of the CBGGP algorithm equipped with the scaled gradient projection operator. In other words, each iteration of the CBGGP consists in solving inexactly the following $K+1$ constrained minimization problems:

$$\begin{aligned} f^{(k+1)} &= \operatorname*{argmin}_{f\in\Omega_f} J_0(f,\omega_1^{(k)},...,\omega_K^{(k)};g,b) \\ \omega_1^{(k+1)} &= \operatorname*{argmin}_{\omega\in\Omega_{\omega_1}} J_0(f^{(k+1)},\omega,...,\omega_K^{(k)};g,b) \\ &\vdots \\ \omega_K^{(k+1)} &= \operatorname*{argmin}_{\omega\in\Omega_{\omega_K}} J_0(f^{(k+1)},\omega_1^{(k+1)},...,\omega_{K-1}^{(k+1)},\omega;g,b) \quad , \end{aligned} \tag{5.34}$$

by means of a given number of SGP iterations. Since each of these subproblems has the form

$$\min_{u\in\Omega} J_0(u)$$

where Ω is one of the above closed and convex sets, the general iteration of SGP will be

$$u^{(k+1)} = u^{(k)} + \lambda_k d^{(k)}$$

where

$$d^{(k)} = P_{\Omega,D_k}(u^{(k)} - \alpha_k D_k^{-1}\nabla J_0(u^{(k)})) - u^{(k)}.$$

As in the previous section, the scaling matrix D_k is the one computed in (5.30) and borrowed from the Lucy-Richardson method, and the steplength α_k is chosen by alternating the two scaled BB rules (5.27). Since the projection operator P_{Ω,D_k} involves a given number of inequalities plus an equality constraint, it can be computed by using the secant-based routine developed in [55], which is able to compute the projection with a computational cost growing linearly in time with respect to the image size.

The problem of blind deconvolution from multiple images finds one of its main application in Fizeau interferometry. As it is known this is a special feature of the Large Binocular Telescope (LBT), which consists of two 8.4 m mirrors situated on a common mount with a center to center distance of 14.4 m. Indeed, this structure is suitable for Fizeau interferometry which should provide images with the resolution of a 22.8 m telescope in the direction of the baseline joining the center of the two mirrors and that of a 8.4 m telescope in the orthogonal direction. It follows that LBT images are characterized by an *anisotropic resolution* and therefore, in order to get the maximum resolution in all directions, it is necessary to acquire different images of the same scientific object with different orientations of the baseline and to combine them into a unique high-resolution image by means of suitable image reconstruction methods. We remark that two interferometers are planned for LBT: the forthcoming LINC-NIRVANA (LN) [83], in advanced realization stage by a German-Italian consortium leaded by MPIA, Heidelberg, and the NASA funded LBTI [142, 10] already operating on Mount Graham.

We now apply Algorithm CBGGP to realistic simulations of imaging both by single mirrors and Fizeau interferometers. On one hand, we consider single image simulations where the PSF is the one of the LBT acquisition system. On the other hand, we generate multiple images by means of the PSFs computed for the interferometer LINC-NIRVANA.

5.3.2 Image simulation

We model the images according to the model proposed in [136] for images acquired with a CCD camera, i.e. each pixel is affected by background (due to sky emission, dark current, etc.), photon counting noise (described by a Poisson distribution) and additive read-out noise (RON) described by a Gaussian distribution.

If the RON variance is σ^2, in the deconvolution process it can be approximated by a Poisson distribution with parameter σ^2 if σ^2 is added both to the detected images and the corresponding backgrounds [136]. Therefore all the pixel values of the detected images can be viewed as realizations of suitable Poisson random variables if in 5.31 we intend that g_j, b_j have been modified according to this approach. Therefore, in our numerical simulations we perturb the images with Poisson and additive Gaussian noise but in the deconvolution algorithms we use the images and backgrounds modified as above.

All the images and the PSFs considered in our numerical experiments are sized 256 × 256 pixels in the single image case, with a pixel size of 15 mas, and 512 × 512 pixels in the multiple image case, with a pixel size of 5 mas. Moreover all images, except one indicated in the sequel, are obtained by adding 10 frames in order to avoid saturation of the detector, as we discuss in the following, so that the variance of the RON will be 10 σ^2.

Single image simulation:
In this case we use two PSFs in K-band with SR = 0.81 and 0.62 respectively, modeling the optics of a single mirror of LBT, with diameter 8.4 m, and the effect of the adaptive optics system FLAO using the power spectrum of the wavefront residual of the AO correction as measured at the telescope [61]. To the noise-free image, obtained by convolving the object with one of these PSFs, a background in K-band is added and the result is corrupted with Poisson and additive Gaussian noise. In order to avoid saturation of the detector (a maximum number of 5×10^4 photons per pixel is assumed in a single frame) the image is obtained by co-adding n frames. More precisely, in the case of a stellar system the procedure for image generation is the following.

- We establish the coordinates of the stars and we fix their magnitudes in K-band.

- We compute the integration time which does not produce saturation of the detector by taking into account the collection area of the telescope, the overall efficiency of the acquisition system (assumed equal to 30%), and the flux of the brightest star multiplied by the peak value of the PSF. This is the integration time of a single frame and is used for computing the number of frames n required for obtaining an acceptable SNR for all the stars of the system.

- We generate noise-free images by shifting, with sub-pixel precision, the PSF to the positions of the stars and adding these shifted PSFs, each one weighted with a weight corresponding to the magnitude and the total observation time.
- These images are perturbed by adding a background in K-band, corresponding to about 13.5 mag arcsec^{-2}, and by corrupting the results with Poisson and additive Gaussian noise (RON); the variance of the RON is $n\sigma^2$, thus corresponding to the RON of n frames; we take $\sigma = 10\ e^-/px$.

Multiple image simulation

As concerns the simulation of LN images, we recall that the instrument combines in a Fizeau mode the beams coming from the two mirrors of LBT whose center-to-center distance is about 14.4 m. Therefore the maximum baseline available is 22.8 m and the resolution achievable by a single LN image is that of a 22.8 m telescope in the direction of the baseline and that of a 8.4 m telescope in the orthogonal direction. For a given orientation the PSF of LN looks as that of a 8.4 m telescope, modulated by the interference fringes, orthogonal to the direction of the baseline. In order to get a more uniform resolution one must acquire and combine different images with different orientations of the baseline.

It is important to remark that the orientation of the fringes does not depend on the orientation of the baseline because the camera is rotating with the baseline and therefore the fringes have always the same direction (for instance the vertical one) in the image array. In other words two images of the same scientific object with two different orientations of the baseline correspond to two rotated versions of that object. This specific feature implies that one should introduce rotation matrices in the formulation of the problem. However we verified that the computation of hundreds or thousands of rotations in hundreds or thousands of inner iterations introduces large computational errors. Therefore we considered the approach which consists in derotating the images in such a way that they correspond to aligned versions of the object f. The price to be payed is that the derotation of discrete images modifies their statistical properties. In order to estimate this effect we considered the rotation of a constant array perturbed by Poisson noise. We found the following results:

- before rotation the histogram of the array is a Gaussian with the same mean and variance; after a rotation based on spline interpolation the histogram is still a Gaussian with the correct mean but a smaller variance;
- the support of the autocorrelation of the rotated image is a 3×3 square;
- if we use a different rotation approach which consists in attributing the value of a pixel before rotation to the pixel with maximum overlapping after rotation (nearest neighbor approximation), the statistics is preserved but the quality of the image is degraded.

As a consequence of this analysis we decided to use in the approach derotated images.

The procedure adopted in our numerical experiments is similar to that used in the case of a single image. We consider two sets of PSFs in K-band with SR respectively 0.77 and 0.46, corresponding to orientation angles of the baseline indicated as 0°, 60° and 120°, all with vertical fringes (for simplicity we take the same SR for the three orientations). The first PSF of each set has been generated by means of the software package LOST [6], the second by reflecting the first one with respect to the central line and the third by taking the arithmetic mean of the first two. In this way the three PSF of each set have exactly the same SR. Then the generation of the corresponding LN images is similar to that of the single image case by modifying the first item as follows.

- We establish the coordinates of the stars corresponding to the observation at 0° and we compute, with sub-pixel precision, their coordinates if the system is rotated by 60° and 120° respectively.

The rest of the procedure is unchanged and applied to the three images but at the end we must add the following item.

- The images corresponding to 60° and 120° are derotated in order to align the object in the three images and three arrays containing the object are extracted from the full images.

The derotated images are used in the definition of the objective function and in the blind algorithm, which therefore will produce derotated PSFs.

5.3.3 Numerical results

In order to evaluate the quality of the reconstructions obtained with our blind method we need some figures of merit.

As concerns the reconstruction of a binary we consider the relative absolute error on the magnitudes of both stars while in the case of a stellar system we consider a magnitude average relative error (MARE) defined by

$$MARE = \frac{1}{q}\sum_{i=1}^{q} \frac{|m_i - \tilde{m}_i|}{\tilde{m}_i} \quad , \tag{5.35}$$

where q is the number of stars and m_i, $\tilde{m}_i$ are respectively the reconstructed and the true magnitudes.

As concerns PSF reconstruction, in the case of single image we consider the root-mean-square error with respect to the true one, defined as usual in terms of the ℓ_2 norm of their difference. In the case of LN images generated according to the previous procedure, since the blind algorithm produces a set of three PSFs, two of them being derotated with respect to the ones used for generating the images, for comparison we must derotate the original ones. If we denote as $\tilde{\omega}_j$ the derotated original PSF, then we measure the quality of the reconstruction by means of the root-mean-square error (RMSE)

$$\rho_j = \frac{\|\omega_j - \tilde{\omega}_j\|}{\|\tilde{\omega}_j\|} \quad , \tag{5.36}$$

where ω_j is the reconstructed PSF and $\| \cdot \|$ denotes the usual ℓ_2-norm.

Binary systems

We first consider the simple case of binary systems. More precisely we consider nine cases by varying both separation and magnitude of the stars. By keeping fixed the magnitude of the primary, i.e. $m_1 = 15$, we take for the magnitude of the secondary $m_2 = 15,\ 16$ and 17. Moreover for each choice we consider three possible angular separations: $d = 60$, 120 and 240 mas in the single image case and $d = 20$, 40 and 80 mas in the LN case. In both cases the first separation corresponds to the resolution limit of the instrument while the last is four times larger. In all cases, as described in the previous section, we compute the integration time of a frame in such a way that the number of counts in the image pixel corresponding to the position of the primary does not exceed 5×10^4. As stated in the previous section, we consider 10 frames per image, both in the single and in the multiple image case, so that the peak value of the photons is about 5×10^5 for all images. Since in the case of LN we have three images, in this case the SNR is higher than in the single image case.

In Figure 5.6 we show the images of the binaries with $m_1 = m_2 = 15$ and different angular separations; in the first row those of the single image case and in the second row those of the multiple image case corresponding to the 0° baseline, all obtained with the PSF with the highest SR. The difficulty in reconstructing the binary with separation $d = 20$ mas is obvious.

Single image

For the convenience of the reader we give the computed integration time avoiding saturation in a single frame: 40 sec for SR = 0.81 and 52 sec for SR = 0.62. As already stated the images are obtained by adding 10 frames. These are the input images of the blind algorithm together with the value of the background.

In a first attempt we use the initialization already used in [113] and in other papers, namely a constant array for the object and the autocorrelation of the diffraction-limited PSF for the PSF. Indeed, this initialization has produced very promising results in our previous paper, where a much higher SNR was assumed. We use 1000 outer iterations in the case SR = 0.81 and 2000 outer iterations in the case SR = 0.62. Indeed in the case of a lower SR we have a lower quality of the images and, presumably, a larger number of iterations is required. As concerns the inner iterations, as in [113] we use 50 SGP iterations for the object and one SGP iteration for the PSF.

In Table 5.3.3 we give the results obtained with the previous choice. As a first remark, the binaries and the PSFs are reconstructed satisfactorily in all cases except the closest binaries

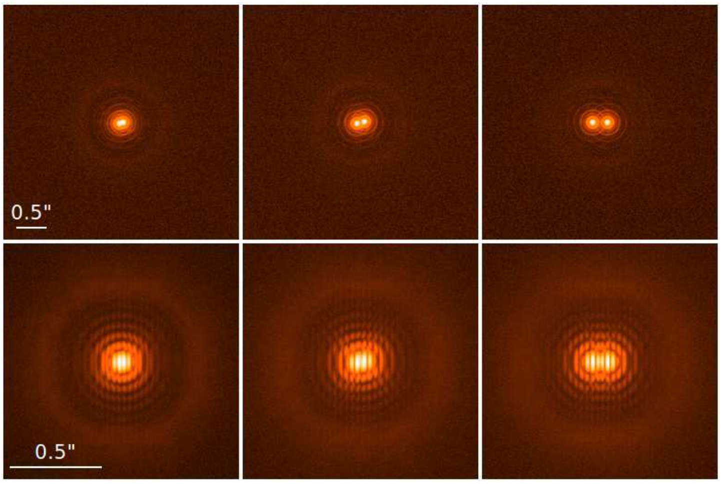


Figure 5.6: Examples of input images of binaries with magnitudes $m_1 = m_2 = 15$. In the first row those of the single image case, corresponding to PSF with SR = 0.81: from left to right, angular separation of 60, 120 and 240 mas. In the second row those of the multiple image case, corresponding to the PSF with SR = 0.77: from left to right, angular separation of 20, 40 and 80 mas. These images correspond to the first orientation of the baseline and only the central part of the images 256 × 256 is displayed. In the two other orientations the binaries appear rotated by 60 and 120 degrees respectively. Images are displayed in log scale. The length corresponding to 0.5 arcsec is also indicated.

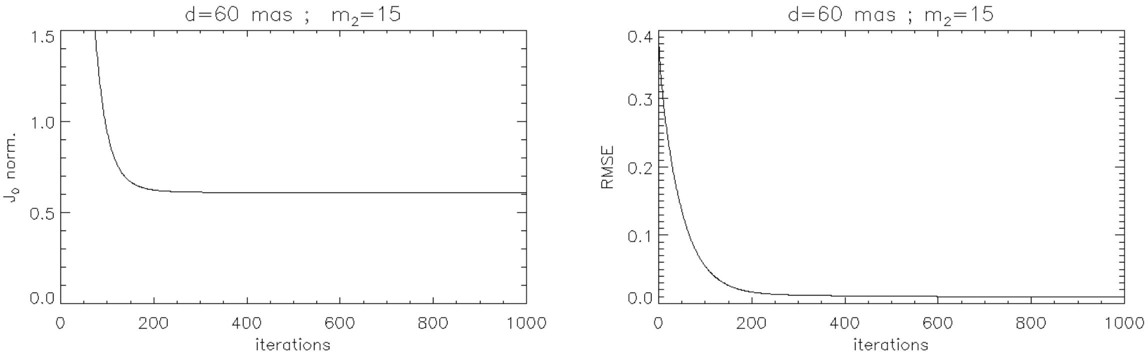


Figure 5.7: Behaviour, as a function of the number of iterations, of the normalized objective function (left panel) and of the RMSE on the PSF (right panel). The parameters of the binary are indicated in the figure. The plots refer to the PSF with SR = 0.81.

($d = 60$ mas) with different magnitudes. Indeed, the indication 100% in the column for $\Delta m_2/m_2$ means that the method reconstructs only one star, which sometimes is not exactly in the position of the primary but slightly shifted in the direction of the secondary. Since its magnitude

SR	d (mas)	m_2	$\Delta m_1/m_1$	$\Delta m_2/m_2$	RMSE	J_0 norm.	IT
0.81	60	15	0.05%	0.03%	0.94%	0.6071	1000
		16	2.37%	100%	40.41%	0.5549	1000
		17	0.99%	100%	16.77%	0.5964	1000
	120	15	<0.01%	<0.01%	0.76%	0.6047	1000
		16	0.03%	<0.01%	1.11%	0.6296	1000
		17	0.03%	0.17%	1.40%	0.6254	1000
	240	15	<0.01%	<0.01%	0.79%	0.5999	1000
		16	0.02%	0.03%	0.83%	0.6273	1000
		17	<0.01%	0.04%	1.17%	0.6229	1000
0.62	60	15	0.18%	<0.01%	1.08%	0.5338	2000
		16	2.31%	100%	34.37%	0.4635	2000
		17	1.05%	100%	16.87%	0.4983	2000
	120	15	0.15%	0.14%	1.04%	0.5261	2000
		16	0.02%	0.01%	1.28%	0.5419	2000
		17	0.04%	0.25%	1.59%	0.5329	2000
	240	15	0.04%	0.04%	1.00%	0.5309	2000
		16	<0.01%	0.06%	1.13%	0.5537	2000
		17	0.05%	0.36%	1.80%	0.5361	2000

Table 5.1: Single image case - Binary reconstructions provided by the algorithm initialized with the autocorrelation of the diffraction-limited PSF. In the first column the value of the SR, in the second the angular separation, in the third the magnitude of the secondary, in the fourth and fifth the errors on the magnitudes of the two stars. In the subsequent column we give the RMSE for the reconstructed PSF. Finally in the last two columns we give the value of the normalized objective function, defined by $2J_0/N^2$, as computed at the end of the iterations, and the number of outer iterations.

is computed using a 3×3 square centered on the true position of the primary, the error on its magnitude is, in general, not too large. On the other hand the error on the PSFs is very large, as one should expect since the secondary is missed. This point deserves further investigation.

In Figure 5.7 we show, in a particular case, the behaviour of the normalized objective function, defined by $2J_0/N^2$ with J_0 given in 5.31 (with $K = 1$), and of the RMSE on the PSF as functions of the number of iterations. Similar behaviors are obtained in all cases where a sensible result is obtained. This result suggests that presumably convergence is reached after 1000 iterations even if it is difficult to establish numerically the convergence of a sequence.

A second remark is that, according to statistical properties of Poisson random variables, if we compute the value of the normalized objective function by inserting in 5.31 the noisy and the noise-free images we should obtain a value very close to 1 [19, 144]. This is just what we obtain using our simulated images (this result also demonstrates the accuracy of the approximation of the RON with a Poisson random variable). However the limiting values of the normalized objective function obtained in our experiments are definitely smaller than 1.

Coming back to the problem of the unresolved binaries, we point out that, if we deconvolve the images using the PSF used for their generation (*inverse crime*) all the binaries are correctly reconstructed with small errors on their magnitudes. Therefore the failure of our experiment may be due to a failure of the method or to an inappropriate initialization or to inappropriate choices of the internal iterations.

Several attempts with different numbers of internal iterations did not improve the results. Therefore we searched for an initial PSF with a SR value closer to the correct one and with the property of being band-limited with the band of the LBT mirror. A possible choice is obtained by means of the diffraction-limited PSF of LBT, let us say $\tilde{\omega}$, by looking for an initial guess $\omega^{(0)}$ of the following form

$$\omega^{(0)} = \frac{1}{1+\nu\ N^2}(\tilde{\omega}+\nu) \tag{5.37}$$

which is band-limited and satisfies the normalization condition. The constant ν should be selected in such a way that $\omega^{(0)}$ has the correct SR value, i.e. max $(\omega^{(0)}) =$ SR max $(\tilde{\omega})$. We obtain

$$\big(\mathrm{SR}\ N^2\ \max(\tilde{\omega})-1\big)\nu = (\mathrm{SR}-1)\ \max(\tilde{\omega}) \tag{5.38}$$

and, by neglecting 1 with respect to the first term in the l.h.s. of this equation, we obtain $\nu = (1-\mathrm{SR})/(\mathrm{SR}\ N^2)$.

The results obtained with this initialization, using again 50 SGP iterations for the object and one for the PSF, are reported in Table 5.3.3. Since the convergence is slower than in the previous case we use 2000 outer iterations for SR = 0.81 and 3000 iterations for SR = 0.62.

By comparing the results reported in the two tables we remark that the two different initializations provide very similar results in all cases where they succeed or they fail; in other words they provide sequences of iterations which presumably converge, even if with a different rate, to the same point, which is a stationary point of the objective function. Obviously we believe that it is also a minimizer. In the case of separation 60 mas and $m_2 = 16$ the algorithm, equipped with the new initialization, is able to reconstruct the binary and the PSF with a satisfactory accuracy for both values of SR. We remark that the value of the objective function is higher than that corresponding to the result provided by the first initialization, which is not correct. This fact clearly indicates the existence of several stationary points or minimizers or both. Of course it should be nice to establish that the result of the first initialization is a stationary point and that of the second a minimizer; but, as already remarked such a verification is practically impossible. Finally, in the case $m_2 = 17$ also the new initialization is unable to provide the correct results.

The results obtained in the multiple image case and described in the next section suggest that this negative result may be due to an insufficient value of the SNR. Therefore, in the case $m_2 = 17$ we generated an image which is the sum of 30 frames (we point out that, as already remarked, in the considered multiple image case we have three times the photons of

SR	d (mas)	m_2	$\Delta m_1/m_1$	$\Delta m_2/m_2$	RMSE	J_0 norm.	IT
0.81	60	15	0.02%	0.04%	0.82%	0.6067	2000
		16	0.09%	0.16%	2.05%	0.6237	2000
		17	1.01%	100%	17.08%	0.5956	2000
	120	15	<0.01%	<0.01%	0.77%	0.6046	2000
		16	0.02%	<0.01%	1.09%	0.6294	2000
		17	0.02%	0.15%	1.35%	0.6253	2000
	240	15	<0.01%	<0.01%	0.80%	0.5989	2000
		16	0.02%	0.02%	0.82%	0.6271	2000
		17	<0.01%	0.02%	1.12%	0.6227	2000
0.62	60	15	0.02%	<0.01%	1.11%	0.5333	3000
		16	0.12%	0.25%	2.64%	0.5354	3000
		17	1.05%	100%	16.87%	0.4983	3000
	120	15	0.01%	0.01%	1.06%	0.5258	3000
		16	0.02%	<0.01%	1.26%	0.5419	3000
		17	0.04%	0.25%	1.58%	0.5329	3000
	240	15	0.03%	0.03%	0.99%	0.5304	3000
		16	<0.01%	0.06%	1.12%	0.5537	3000
		17	0.05%	0.36%	1.80%	0.5361	3000

Table 5.2: Single image case - Binary reconstructions provided by the algorithm initialized with the diffraction-limited PSF plus a constant selected for satisfying the SR constraint (see the text). The structure of the Table is the same of Table 5.3.3.

the single image case). Using again 2000 iterations, we find that the algorithm, with the second initialization, can resolve the binary in the case SR = 0.81 (even if with a large reconstruction error, about 9 %, on the PSF) but not in the case SR = 0.62.

However in these difficult cases we observe a new phenomenon: even if in the limit the results are not satisfactory, the PSF reconstruction error exhibits a minimum before convergence. If we consider the reconstructions corresponding to these minima, then, in the case of the first initialization, the minima do not correspond to a situation where the binary is resolved. On the other hand, in the case of the second initialization, the binary is resolved for both SR values, with a 2.03 % PSF error in the case SR = 0.81 (574 iterations) and a 7.13 % error in the case SR = 0.62 (1739 iterations). Such a result presumably indicates the need of introducing a regularization of the PSF in the objective function, at least for treating the most difficult cases. In Figure 5.8 we show the reconstructions of the PSF corresponding to the minimum reconstruction errors. Artifacts due to the missed secondary are visible in the case of the first initialization and also in the case SR = 0.62, since the reconstructed secondary is fainter than the true one.

Multiple images

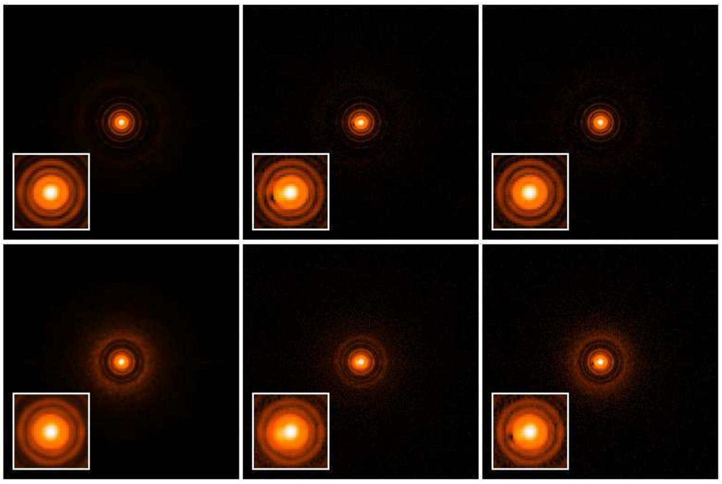


Figure 5.8: Single image case - PSF reconstruction in the case of the binary with $d = 60$ mas and $m_2 = 17$. The input image is the sum of 30 frames (see text). These PSFs correspond to the minima of the reconstruction error. First column: the true PSF with SR $=$ 0.81 (top) and SR $=$ 0.62 (bottom). Second column: PSF reconstruction provided by the algorithm initialized with the autocorrelation of the diffraction-limited PSF. Last column: PSF reconstruction provided by the algorithm initialized with the diffraction-limited PSF plus a constant. In each panel we also show a zoom of the core of the PSF which makes evident artifacts due to the secondary. All images are displayed in log scale.

In this case the integration time of a nonsaturated frame is 95 sec for SR $= 0.77$ and 167 sec for SR $= 0.46$. For each binary and orientation angle we consider again 10 frames, so that we have approximately the same number of photons in all images.

We preliminarily remark that, if we compute the value of the normalized objective function (which is now given by $2J_0/3N^2$) by inserting in 5.31 the noisy and the noise-free images before derotation, we expect to obtain a value very close to 1 and this is just what we obtain. But this is not true if we compute the same quantity using the derotated images. Indeed, for the nine binaries as well as for the other objects, we always obtain a smaller value, namely 0.63. Since this value is independent of the object and PSFs, this effect is clearly due to the modification of the statistical properties of the data introduced by the derotation, as briefly discussed in Section 5.3.2. In any case the limiting values of the normalized objective function obtained in our experiments are definitely smaller than the values corresponding to the input objects and images, an effect already remarked in the previous case.

As in the single image case we first use as initialization a constant array for the object and the autocorrelations of the ideal PSFs for the three PSFs. The results of the reconstructions obtained with this initialization are reported in Table 5.3.3. We obtain that only when both stars have the same magnitude the method is able to reconstruct both the binary and the PSFs

SR	d (mas)	m_2	$\Delta m_1/m_1$	$\Delta m_2/m_2$	RMSE$_{0^\circ}$	RMSE$_{60^\circ}$	RMSE$_{120^\circ}$	J_0 norm.	IT
0.77	20	15	0.28%	0.23%	1.52%	2.50%	1.87%	0.2241	1000
		16	2.12%	100%	30.00%	31.14%	21.79%	0.1706	1000
		17	0.87%	100%	14.93%	15.38%	11.00%	0.1890	1000
	40	15	0.30%	0.28%	1.66%	1.84%	2.56%	0.2647	1000
		16	2.20%	100%	41.65%	33.98%	33.80%	0.2049	1000
		17	0.58%	100%	14.78%	14.48%	15.90%	0.1954	1000
	80	15	0.20%	0.21%	1.09%	0.83%	0.83%	0.2180	1000
		16	0.18%	0.23%	1.27%	0.96%	0.99%	0.2164	1000
		17	0.87%	100%	19.28%	19.06%	19.04%	0.1893	1000
0.46	20	15	1.27%	1.19%	9.67%	9.69%	11.39%	0.1335	1000
		16	0.29%	100%	32.61%	33.10%	29.37%	0.0836	1000
		17	0.18%	100%	13.91%	14.12%	12.54%	0.0795	1000
	40	15	0.89%	0.88%	5.15%	5.62%	5.64%	0.1516	1000
		16	0.27%	100%	47.22%	41.54%	36.75%	0.1360	1000
		17	0.33%	100%	13.91%	14.67%	14.31%	0.0944	1000
	80	15	0.68%	0.68%	3.05%	2.60%	2.58%	0.1042	1000
		16	0.52%	0.60%	1.87%	1.43%	1.44%	0.0850	1000
		17	0.55%	100%	15.99%	15.97%	15.98%	0.0883	1000

Table 5.3: Multiple image case - Binary reconstructions provided by the algorithm initialized with the autocorrelations of the ideal PSFs. In the first column the value of the SR, in the second the angular separation, in the third the magnitude of the secondary, in the fourth and fifth the errors on the magnitudes of the primary and the secondary star. In the subsequent three columns we give the RMSE for the three PSFs. Finally in the last two columns we give the value of the normalized objective function, defined by $2J_0/3N^2$, as computed at the end of the iterations, and the number of outer iterations.

with sufficient accuracy. When we have different magnitudes for the two stars the method is in general failing to reproduce the secondary, except in the case of separation $d = 80$ mas; in this case a binary with difference of magnitude $\Delta m = 1$ is also reconstructed. As in the single image case, the indication 100% in the column for $\Delta m_2/m_2$ means that the method reconstructs an object which contains only one bright star (in one case the centroid is shifted one pixel in the direction of the secondary. These results show that, even if we have a higher SNR as already discussed, the multiple image case is more difficult than the single one.

If we deconvolve the derotated images using the derotated PSFs (this is not exactly an *inverse crime* because the images were generated with non derotated PSFs) all the binaries are correctly reconstructed with small errors on the magnitudes. Therefore the failure of our experiment may be due again to an inappropriate initialization (the autocorrelations of the ideal PSFs have a SR value of about 0.35, much smaller than the SR of the PSFs used in image

SR	d (mas)	m_2	$\Delta m_1/m_1$	$\Delta m_2/m_2$	RMSE$_{0^\circ}$	RMSE$_{60^\circ}$	RMSE$_{120^\circ}$	J_0 norm.	IT
0.77	20	15	0.44%	0.34%	2.86%	4.23%	3.42%	0.2277	2000
		16	0.27%	0.21%	1.56%	1.82%	1.74%	0.2209	2000
		17	0.07%	1.10%	2.53%	2.70%	1.78%	0.2095	2000
	40	15	0.45%	1.03%	5.47%	4.61%	6.73%	0.2670	2000
		16	0.25%	0.39%	1.63%	2.85%	2.76%	0.2220	2000
		17	0.11%	0.73%	2.05%	2.78%	2.86%	0.2102	2000
	80	15	0.35%	0.35%	2.28%	1.51%	2.23%	0.2204	2000
		16	0.25%	0.26%	1.32%	1.06%	1.14%	0.2179	2000
		17	0.19%	0.40%	1.32%	1.02%	0.99%	0.2125	2000
0.46	20	15	0.80%	0.57%	4.02%	4.51%	6.83%	0.1037	2000
		16	0.38%	0.95%	3.76%	3.95%	2.52%	0.0811	2000
		17	0.07%	6.32%	9.05%	10.33%	6.29%	0.0697	2000
	40	15	0.64%	2.18%	11.23%	6.72%	8.73%	0.1409	2000
		16	0.49%	0.81%	2.21%	3.20%	2.99%	0.0837	2000
		17	0.02%	5.89%	8.70%	7.68%	8.84%	0.0716	2000
	80	15	0.56%	0.55%	8.54%	4.66%	4.64%	0.1100	2000
		16	0.57%	0.53%	1.99%	1.48%	1.49%	0.0846	2000
		17	0.48%	0.92%	2.36%	1.95%	1.96%	0.0785	2000

Table 5.4: Multiple image case - Binary reconstructions provided by the algorithm initialized with the ideal PSFs plus a constant selected for satisfying the SR constraint (see the text). The structure of the Table is the same of Table 5.3.3.

generation) or to inappropriate choices of the internal iterations. Also in this case, as in [113] and in the single image case, we use 50 SGP iterations for the object and one SGP iteration for each PSF. However several attempts with different numbers of internal iterations did not improve the results. Therefore, as in the single image case, we use as a new initialization of the PSFs the ideal PSFs of LN with the addition of a small constant selected in such a way to satisfy normalization and SR value. The results obtained with this initialization, using again 50 SGP iterations for the object and one for the PSFs, are reported in Table 5.3.3. Since the convergence is slower than in the previous case we use 2000 outer iterations.

With the new initialization the blind method succeeds in reconstructing all the binaries with sufficient accuracy as well as the PSFs. We can add that in most cases both the normalized objective function and the RMSE on the PSFs have a convergent behaviour while, in a few cases, the errors are still decreasing after 2000 iterations, thus indicating that a larger number of iterations could still improve the solution. A comparison of the values of the objective function reported in the two tables shows that, in some of the cases where the first initialization is failing, the values in Table 5.3.3 are smaller than the corresponding values in Table 5.3.3. This phenomenon was already observed in the single image case and means that different stationary

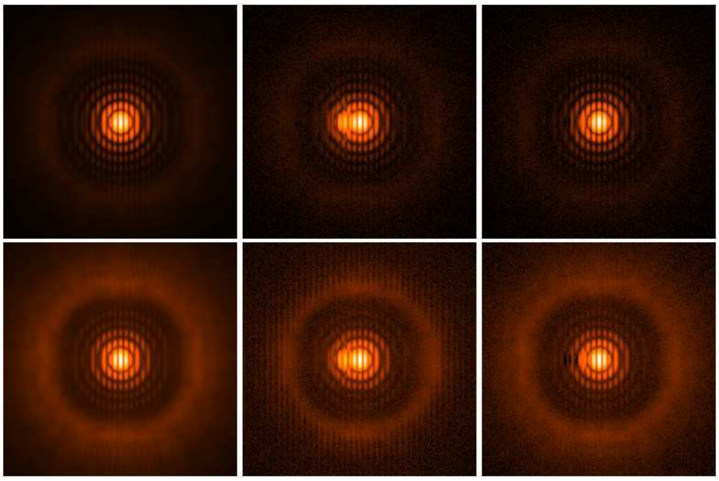


Figure 5.9: Multiple image case - PSF reconstruction in the case of the binary with $d =$ 80 mas and $m_2 = 17$. First column: the true PSF with SR = 0.77 (top) and SR = 0.46 (bottom). Second column: PSF reconstruction provided by the algorithm initialized with the autocorrelations of the ideal PSFs. Last column: PSF reconstruction if the algorithm is initialized with the ideal PSFs plus a constant. All images (only the central part 256×256 is shown) are displayed in log scale and correspond to the first orientation of the baseline.

points or minimizers are present.

A few more comments on the two tables. If one looks carefully at the reported results one can remark that, even if the results obtained with the second initialization are globally better than those obtained with the first one, this may not be true for particular cases (compare, for instance, the results for $d = 40$ mas and $\Delta m = 0$). Moreover, the errors obtained with the second initialization do not vary in a regular way with the variation of angular distance and difference of magnitude. These behaviors can be due to the fact that 2000 iterations may not be sufficient for assuring convergence of the method in the case of the second initialization. We did not push further the iterations because in the case of three 512×512 images the computation time is considerable. By assuming possible fluctuations due to insufficient number of iterations, a reasonable conclusion seems to be that, as in the single image case, the two initializations lead to the same limit point when the first one is successful.

In Figure 5.9 we show an example of reconstructions of the PSF at 0°, for both SR values, when the unknown object is a binary with $d = 80$ mas and $m_2 = 17$. From the reconstructions displayed in the second column and obtained by initializing with the autocorrelations of the ideal PSFs, it is evident that they contain a contribution coming from the secondary, while this contribution is practically absent in the reconstructions obtained with the other initialization and displayed in the third column.

Star clusters

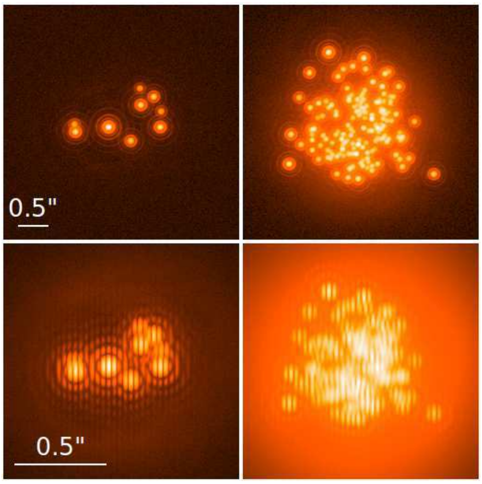


Figure 5.10: Top panels: the input images of the "open star cluster" (left) and of the "globular star cluster" (right) in the single image case with SR = 0.81. Bottom panels: the input images of the two clusters in the case of SR = 0.77 and with 0° of the baseline (only the central part 256×256 is shown). All images are displayed in log scale. The length corresponding to 0.5 arcsec is also indicated.

In a second experiment we consider two models of star cluster. The first is already considered in [113] and is based on an image of the brightest stars of the Pleiades open cluster; for this reason, we call it "open star cluster". It consists of nine stars that we take, in this paper, with magnitudes ranging from 14.4 to 17.1. In the single image case, the minimum distance between two stars is 120 mas, while the maximum distance is 1434 mas, with a mean distance of about 690 mas. In the multiple image case, considering the different pixel scale, we reduce of one third all the distances.

As a second example we consider a model that we call "globular star cluster". For simplicity, only 150 stars are considered within the field of view, representing a very low crowding condition. The positions of the stars are randomly computed following a Gaussian distribution around the center of the image (with a standard deviation of about 450 mas in the single image case and of about 150 mas in the multiple image case); similarly the magnitudes of the stars are randomly distributed around $m = 16$ with a standard deviation of about 0.4. It turns out that the brightest star of the cluster has $m = 14.8$.

Again, we limit the maximum number of counts in each frame to 5×10^4, keeping fixed to 10 the number of frames. In Figure 5.10 we show the images of the two star clusters provided by the PSFs with the highest SR.

Single image

In the case of the "open star cluster", the integration time of a single frame is 22 sec for SR = 0.81 and 29 sec for SR = 0.62 while in the case of the "globular star cluster" these times are

Star Cluster	SR	Init.	MARE	RMSE	J_0 norm.	IT
OC	0.81	A	0.06%	0.84%	0.5993	2000
		C	0.06%	0.85%	0.5991	5000
	0.62	A	0.09%	1.22%	0.5165	4000
		C	0.09%	1.22%	0.5165	10000
GC	0.81	A	0.06%	0.87%	0.5123	3000
		C	0.06%	0.82%	0.4989	5000
	0.62	A	0.07%	1.07%	0.4622	6000
		C	0.07%	15.82%	0.5698	10000

Table 5.5: Single image case - The reconstruction errors in the case of the two star cluster models. In the first column the "open cluster" is labelled by OC, while the "globular cluster" is GC. In the second column, we give the value of SR, while in the third column we give the initialization of the algorithm, denoting by A the autocorrelation of the diffraction-limited PSF and by C the diffraction-limited PSF plus a constant selected for satisfying the SR constraint (see the text). In the subsequent columns, we give the value of the magnitude average reconstruction error (MARE) defined in 5.35, and the RMSE for the reconstructed PSF. Finally in the last two columns we give the value of the normalized objective function, defined by $2J_0/N^2$, as computed at the end of the iterations, and the number of outer iterations.

respectively 32 and 42 sec.

We applied to the four images our blind algorithm using both initializations introduced in the case of the binaries. The results are reported in Table 5.3.3. In the case of the "open star cluster" and both values of SR the two initializations seem to provide sequences of iterations converging to the same point. If we look at the image shown in the upper left panel of Figure 5.10 we can observe that it contains sufficiently well-separated star images which can allow a good estimation of the PSF by the blind algorithm.

The situation is a bit different in the case of the "globular star cluster" and we can understand this fact if we look at the upper right panel of Figure 5.10. In the case of the higher SR value both initializations lead essentially to the same result. The small differences may be due to different convergence rates and could be removed by a more accurate tuning of the number of iterations. On the other hand in the case of the lower SR ratio the first initialization, based on the autocorrelation of the diffraction-limited PSF, provides the best PSF reconstruction (also corresponding to a lower value of the objective function). It seems that the two initializations lead to two different stationary points. In conclusion, for this particular object one can state that the first initialization may provide a better result than the second one.

Multiple images

In the case of the "open cluster" model, the integration time is 53 sec for SR = 0.77 and 93 sec for SR = 0.46. On the other hand the integration time for the "globular cluster" images is 78 sec for SR = 0.77 and 136.5 sec for SR = 0.46.

Star Cluster	SR	Init.	MARE	RMSE$_{0^\circ}$	RMSE$_{60^\circ}$	RMSE$_{120^\circ}$	J_0 norm.	IT
OC	0.77	A	0.35%	2.14%	4.28%	4.19%	0.3049	1000
		C	0.59%	4.16%	7.67%	7.62%	0.2997	5000
	0.46	A	0.81%	3.43%	6.07%	6.00%	0.1321	2000
		C	0.89%	3.74%	7.77%	7.81%	0.1237	10000
GC	0.77	A	0.38%	1.98%	3.46%	3.38%	0.7597	3000
		C	0.71%	11.00%	11.06%	12.97%	0.7459	5000
	0.46	A	1.04%	5.63%	9.69%	9.38%	0.3557	6000
		C	0.90%	25.81%	16.37%	17.94%	0.3043	10000

Table 5.6: Multiple image case - The reconstruction errors in the case of the two models of star cluster. The structure is similar to that of Table 5.3.3 but now we give the errors on the three PSFs and the normalized objective function is defined by $2J_0/3N^2$.

In both cases we apply our blind algorithm using the two initializations already used in the previous sections, with 50 inner SGP iterations for the object and one iteration for each PSF. The results obtained for the "open cluster" with the two initializations are given in the first two rows of Table 5.3.3 in the case SR = 0.77 and in the following two rows those obtained in the case SR = 0.46. Similarly the results obtained for the "globular cluster" are given in the second half of the same table.

In the multiple image case the situation is more complex than in the single one, and this is not surprising since now we must reconstruct four blocks of variables. By looking at the results reported in Table 5.3.3, we see that the two initializations produce in all cases two sequences of iterations converging to distinct results. Even if, in some cases, the two values of the objective function are very close, the corresponding points are definitely different, thus implying the existence of several minimizers or stationary points with very close values of the objective functions.

It is interesting to remark that, while in the case of the binaries the best results are provided by the second initialization, now they are provided by the first one, based on the autocorrelations of the ideal PSFs. The highest reconstruction errors are obtained in the case of the lowest SR, as one should expect. We also remark that in the case of the second initialization we used a larger number of iterations because the convergence is slower than in the case of the first initialization. From the comparison of the results obtained for the binaries with those obtained for the star clusters we deduce that the problem of the initial PSFs is essentially open; therefore, in the case of practical applications, one should try with different initializations, using also physical intuition in their choice.

As a final comment, all the values of the objective function corresponding to the best solutions are higher than those corresponding to the other ones.

Conclusions

The development of efficient first order methods for nonlinear optimization is of great interest in several scientific applications, thanks to their simplicity and low computational cost per iteration. However, these methods need acceleration strategies in order to be computationally efficient and competitive with other non-iterative approaches. This thesis gave a contribution to this issue by presenting variable metric line–search based methods suited for a particular class of optimization problems, in which the objective function is given by the sum of a differentiable, possibly nonconvex term and a convex, possibly nondifferentiable term.

First, we proposed the proximal–gradient method VMILAn, in which the free choice of the parameters defining the metric of the proximal operator is combined with an Armijo-like condition to ensure the convergence of the scheme. Notably, the general VMILAn algorithm defines the proximal step with an inexactness criterion, in order to take into account the case in which the proximal operator cannot be computed explicitly. This criterion is practically implementable when the convex part is given by the composition of a proper, convex and continuous term with a linear operator, which allows to include the almost totality of the regularization terms adopted, for instance, in the context of image processing.

Second, we devised an inexact version of the nonlinear Gauss-Seidel scheme for the minimization of a differentiable objective function subject to separable constraints, which is recovered in the general class of problems above mentioned by choosing as the convex term the sum of indicators of closed, convex sets. More in detail, we address this problem by performing inexactly the minimization by means of a fixed number of the gradient projection steps, where the projection may be computed with respect to non Euclidean metrics. Special instances of these metric are the scaled Euclidean and Bregman distances.

For both methods, the general result of the stationarity of the limit points is proved without convexity assumptions. In the case of VMILAn, strong convergence of the iterates to a stationary point is also proved when the objective function satisfies the Kurdyka–Łojasiewicz property. As we have seen in Section 2.3.1 of Chapter 2, this is a rather general and not restrictive assumption, satisfied by the majority of data fidelity functions and regularization terms used in signal and image processing.

Extensive numerical experience in image processing applications, such as image deblurring and denoising in presence of non-Gaussian noise, image compression, phase estimation in

DIC microscopy and image blind deconvolution, has shown the flexibility of our methods in addressing different nonconvex problems, as well as their ability to effectively accelerate the progress towards the solution of the treated problem with respect to other comparable approaches proposed in the literature. We remark that the parameters involved are chosen by adaptive strategies well known in the literature, such as the alternation of the Barzilai Borwein rules for the steplengths, and the Split Gradient or Majorize-Minimize techniques for the scaling matrices. In our opinion, the variable choice of both parameters is what triggers off the acceleration behaviour shown by our methods. Indeed, the combination of adaptive strategies for both the steplength and scaling matrix seems to lead to the best improvement in terms of numerical efficiency in the majority of cases. There are of course exceptions: as an example, the best choice in the image reconstruction test in Section 3.3.1 is provided by coupling the identity matrix with an adaptive steplength selection based on a Lanczos-like process. In general, we can say that the proposed methods always benefit from the variable choice of at least one of the involved parameters. Another major strength of the methods is robustness with respect to the noise level on the data: this is observed in Section 4.3 of Chapter 4, where the non scaled version of VMILAn recovers good reconstructions of the phase function even for high values of the signal-to-noise ratio.

Future work will concern a systematic and rigorous treatment of the selection of the parameters in the proposed methods, with a particular attention for VMILAn. In this case, we will investigate on how the parameters selection affects the inner subproblem for the computation of the inexact proximal point, as well as understanding how the approximation level of the proximal point influences the algorithmic performances. Another subject of future research will be the combination of VMILAn to the inexact Gauss-Seidel scheme proposed in Chapter 5, with the consequent application of the resulting scheme in the context of image blind deconvolution in astronomical imaging, by adding nondifferentiable regularization terms suited to the reconstruction of diffuse objects.

Appendix A

Basics on image restoration

The goal of image restoration is to restore the original image x from a degraded acquired image y. Usually, one can roughly distinguish between two kinds of degradation:

- the degradation due to the process of image formation, which is denominated *blurring*;
- the degradation introduced by the recording process of the image, which is called *noise* and is triggered off by measurement or counting errors.

Mathematical modeling of image acquisition
Image reconstruction techniques are based on an *image formation model*, which describes the propagation of the radiation used in the imaging process. If we denote with $x(s)$ a function of the space variables describing the unknown *object* and with $\bar{y}(s)$ the acquired *noise-free image*, then the optical image formation is frequently modelled by the following continuous linear model [17]

$$\bar{y}(s) = \int \mathcal{H}(s, s')x(s')ds' \tag{A.1}$$

where $\mathcal{H}(s, s')$ is the *Point Spread Function (PSF)*. The term comes from the fact that $\mathcal{H}(\cdot, s')$ is the image of a point source located at the point s'. Indeed, if the object is given by $u(s'') = \delta(s'' - s')$, where $\delta(\cdot)$ indicates the Delta distribution, then according to (A.1), one obtains $\bar{y}(s) = \mathcal{H}(s, s')$. The effect of the PSF is called *blurring* and $\bar{y}$ is the blurred image.

In several acquisition systems, the PSF is assumed to be *space-invariant*, i.e. invariant with respect to translations; in this case, the function $\mathcal{H}(s, s')$ depends only on the difference $s - s'$ and model (A.1) reduces to

$$\bar{y}(s) = \int \mathcal{H}(s - s')x(s')ds' = (\mathcal{H} \otimes x)(s) \tag{A.2}$$

where $\otimes$ denotes the convolution product.

Furthermore, when images are treated as digital signals, a discrete version of model (A.2) is required. In this case, the unknown object and the PSF will be two vectors $x, h \in \mathbb{R}^n$, and the convolution product can be seen as the product matrix-vector Hx, where $H \in \mathbb{R}^{m\times n}$ is the convolution matrix obtained by imposing some specific boundary conditions on the discretized PSF h [81]. Finally, taking into account the presence of noise in the recording process and adding also a nonnegative constant *background term* b to the model, we can write

$$y = Hx + b\boldsymbol{e} + v \tag{A.3}$$

where $y \in \mathbb{R}^m$ is the blurred and noisy image and $v \in \mathbb{R}^m$ represents the additive noise contribution.

Concerning the matrix H, standard assumptions are the following:

$$H_{i,j} \geq 0,\ \forall\, i,j, \quad H^T\boldsymbol{e} = \boldsymbol{e}, \quad H\boldsymbol{e} > 0.$$

Let us also remark that, if *periodic* boundary conditions are employed in model (A.3), i.e. if the two-dimensional PSF $h = \{h_{i,j}\}_{i=1,\ldots,m}^{j=1,\ldots,n}$ is such that

$$h_{m+1,j} = h_{1,j}, \quad h_{i,m+1} = h_{i,1}, \quad \forall\, i,j \tag{A.4}$$

then H is block circulant with circulant blocks and the matrix-vector products Hx and H^Tx can be efficiently computed by making use of the Discrete Fourier Transform (DFT) and its inverse (IDFT) [17, 81]. Indeed, by means of the convolution theorem, we have

$$\begin{aligned} Hx &= \mathrm{IDFT}\left(\mathrm{DFT}(h)\cdot\mathrm{DFT}(x)\right)\\ H^Tx &= \mathrm{IDFT}\left(\overline{\mathrm{DFT}(h)}\cdot\mathrm{DFT}(x)\right), \end{aligned}$$

where $\overline{\alpha}$ denotes the complex conjugate of $\alpha \in \mathbb{C}$. Hence, the above matrix-vector products may be performed with a $\mathcal{O}(mn\log(mn))$ complexity. This efficient computation is guaranteed also with other boundary conditions, such as zero or reflexive conditions, which imply the use of other discrete transforms apart from the DFT [81].

The noise vector v in model (A.3) can be seen as a realization of a random variable and, as a consequence, each pixel y_i of the acquired image can be seen as a realization of a random variable Y_i. By setting $Y = (Y_1,\ldots,Y_m)$, the modelling of the system is then related to the probability density of the multivariate random variable Y. This density depends on the object x and therefore we denote it as $p_Y(y;x)$. The following assumptions on Y_i and Y are usually accepted as reasonable ones [17, 20]:

- the random variables Y_i are statistically independent, that is
$$p_Y(y;x) = \prod_{i=1}^{m} p_{Y_i}(y_i;x);$$
- the expected value of Y_i is given by the $i-$th pixel of the noise-free image, hence
$$E(Y) = \int y p_Y(y;x)dy = Hx + b\boldsymbol{e}.$$

There are then two classical examples of noise modelling:

- **White Gaussian noise:** each component v_i of the noise vector v is a realization of a random variable with Gaussian distribution of zero mean and standard deviation $\sigma > 0$. Then the vector v is a realization of the multivariate random variable V, whose probability density is
$$p_V(v) = \frac{1}{(2\pi\sigma^2)^{m/2}} \exp\left(-\frac{1}{2\sigma^2}\|v\|^2\right).$$
Therefore the statistical model for the detected image is
$$p_Y(y;x) = \frac{1}{(2\pi\sigma^2)^{m/2}} \exp\left(-\frac{1}{2\sigma^2}\|y-(Hx+b)\|^2\right). \tag{A.5}$$

- **Poisson noise:** each Y_i is a Poisson random variable with expected value given by $(Hx+b\boldsymbol{e})_i$. By invoking the statistical independence of the random variables Y_i, we have
$$p_Y(y;x) = \prod_{i=1}^{m} \frac{e^{-(Hx+b\boldsymbol{e})_i}(Hx+b\boldsymbol{e})_i^{y_i}}{y_i!}. \tag{A.6}$$

Maximum likelihood (ML) approach

In image restoration, one wants to recover the object x corresponding to the image y: this is an example of *inverse problem* [78, 17]. A naive approach to address this problem is to compute

$$x = H^{-1}(y - b\boldsymbol{e})$$

as the solution of the linear system $Hx = y - b\boldsymbol{e}$. However, this approach is not suitable when the matrix H is not invertible, i.e. when the problem is *ill posed*, or when H has a high condition number, i.e. when the problem is *ill-conditioned*. Of course there are cases in which direct inversion of the linear model (A.3) is practicable, such as computed tomography, but these are only exceptions.

Since we assume that the probability density $p_Y(y;x)$ is known, it is then natural to look for statistical formulations of the image restoration problem. The standard approach is the so-called *maximum likelihood* (ML) estimation [135], in which an estimate of the unknown object x is any x^* that maximizes the probability density of y, denominated the *likelihood function* of the problem:
$$x^* = \underset{x\in\mathbb{R}^n}{\operatorname{argmax}}\ p_Y(y;x).$$

Clearly, this is equivalent to minimize the negative logarithm of the probability density. Therefore, the ML problem may be reformulated in the following alternative way:

$$x^* = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f_0(x;y) \equiv -A\ln(p(y;x)) + B \tag{A.7}$$

where A and B are suitable real constants. The function f_0 is referred to as the *fit-to-data term*, since it measures the distance between the observed data and the one predicted by the linear model.
Different noise models lead to different functionals f_0. In particular:

- **Gaussian noise:** setting $A = \sigma^2$ and $B = A/(2\pi\sigma^2)^{m/2}$, we have

$$f_0(x; y) = \frac{1}{2}\|Hx + b\boldsymbol{e} - y\|^2 \tag{A.8}$$

  which leads to the classical *Least Squares* (LS) minimization problem.

- **Poisson noise:** using Stirling's formula to approximate the factorial and neglecting some constants, we obtain

$$f_0(x; y) = \mathrm{KL}(Hx + b\boldsymbol{e}; y) \tag{A.9}$$

  where

$$\mathrm{KL}(x; y) = \sum_{x_i > 0} y_i \log\left(\frac{y_i}{x_i}\right) + x_i - y_i$$

  is the Kullback–Leibler functional.

For both types of noise, it can be seen that problem (A.7) is still affected by the possible ill-conditioning of the matrix H [17]. This means that one should not aim at computing the minimum points of the functional f_0, since they do not provide sensible estimates of the unknown object. In this sense, very efficient methods, such as second order methods, pointing directly to the minima, can be dangerous. On the other hand, first order methods can provide acceptable (regularized) solutions by early stopping.

Maximum A Posteriori (MAP) approach

A more complete statistical framework is provided by the *Bayesian approach* [70], in which we assume that the unknown object x is also a realization of a multivariate random variable X. The probability density of X is the so-called *prior* and will be denoted by $p_X(x)$. Introducing also the marginal probability $p_Y(y)$, we can compute, by means of the *Bayes theorem*, the conditional probability of X with respect to the given value y of Y:

$$p_X(x|y) = \frac{p_Y(y|x)p_X(x)}{p_Y(y)}.$$

In this manner, some properties of the object (such as smoothness, sharp edges etc) can be incorporated in the *a priori* probability $p_X(x)$. The most frequently used priors are of the Gibbs type:

$$p_X(x) = c\exp\left(-\lambda f_1(x)\right)$$

where $c \in \mathbb{R}$, $\lambda \in \mathbb{R}_{>0}$, and f_1 is a functional which is usually convex.

Then, a *maximum a posteriori* (MAP) estimate of the unknown object is any x^* that maximizes the a posteriori probability $p_X(x|y)$:

$$x^* = \operatorname*{argmax}_{x \in \mathbb{R}^n} p_X(x|y).$$

By consider the equivalent formulation with the negative logarithm of $p_X(x|y)$ and assuming that $p_X(x)$ is a Gibbs prior, we have:

$$\begin{aligned} x^* &= \operatorname*{argmin}_{x \in \mathbb{R}^n} - \ln p_X(x|y) = \operatorname*{argmin}_{x \in \mathbb{R}^n} \left(- \ln(p_Y(y|x)) - \ln(p_X(x)) + \ln(p_Y(y))\right) \\ &= \operatorname*{argmin}_{x \in \mathbb{R}^n} f(x;y) \equiv f_0(x;y) + \lambda f_1(x) \end{aligned}$$

The function f_1 is called *regularization functional*, and has the role of imposing some properties on the desired solution, whereas λ is the *regularization parameter*, which balances the trade-off between f_0 and f_1. In presence of Gaussian or Poisson noise, the function f_0 is differentiable, coercive and convex; hence, if f_1 is also a convex function, then f admits global minimizers for any positive value of λ. However, it is worth noting that the quality of the reconstructions obtained via a MAP approach hugely depends on the choice of the parameter λ.

Regularization functionals

In the following, the symbol ∇ denotes the discrete gradient operator, i.e. $\nabla = (\nabla_1^T, \dots, \nabla_n^T)^T$ where $\nabla_i \in \mathbb{R}^{2 \times n}$ operates the forward finite differences at the $i-$th pixel of the image:

$$\nabla_i x = \begin{pmatrix} x_{i+1} - x_i \\ x_{i+m} - x_i \end{pmatrix}$$

where $x \in \mathbb{R}^n$ represents a vectorized 2D image. A similar definition is given for the Laplacian operator ∇^2.

Let us recall some of the most classical regularizers used in image deblurring and denoising.

- **Tikhonov regularization:** given $A \in \mathbb{R}^{n \times n}$, the choice

  $$f_1(x) = \frac{1}{2} \|Ax\|^2$$

  is known as *Tikhonov regularization* [138], and its aim is to emphasize smooth details in the object. According to the choice of A, we distinguish between zero-order, first-order and second-order Tikhonov regularization:

  - $A = I$ (zero-order);
  - $A = \nabla$ (first-order);
  - $A = \nabla^2$ (second-order).

  All these functionals are continuously differentiable and convex.

- **Edge-preserving regularization:** in contrast with the Tikhonov regularizers, the *Total Variation* (TV) functional [132] preserves discontinuities and edges in the image. The discrete version of Total Variation can be written as

  $$TV(x) = \sum_{i=1}^{n} \|\nabla_i x\| = \sqrt{(x_{i+1} - x_i)^2 + (x_{i+m} - x_i)^2}.$$

  Note that the functional $TV(x)$ is convex; however, it is nondifferentiable at any point x such that $\nabla_i x = 0$ for some $i \in \{1, \ldots, n\}$. To avoid such points and recover differentiability, one can introduce a positive threshold $\delta \in \mathbb{R}_{>0}$ and consider the *Hypersurface Potential* (HS) functional [141, 43]:

  $$HS(x) = \sum_{i=1}^{n} \sqrt{\|\nabla_i x\|^2 + \delta^2} = \sqrt{(x_{i+1} - x_i)^2 + (x_{i+m} - x_i)^2 + \delta^2}.$$

Constraints

In some applications, a priori information comes also from the physics underlying the acquisition process. Additional information of this kind may be added by restricting the search of the object x onto a convex set Ω:

$$x^* = \operatorname*{argmin}_{x \in \mathbb{R}^n} f_0(x; y) + f_1(x) + \iota_\Omega(x).$$

Among the most typical examples of constraint sets, we recall:

- the nonnegative orthant: $\Omega = \mathbb{R}^n_{\geq 0}$, used to impose the nonnegativity on the image pixels;
- conservation of the flux: $\Omega = \{x \in \mathbb{R}^n : \ \sum_{i=1}^{n} x_i = c\}$;
- box constraint: $\Omega = \{x \in \mathbb{R}^n : \ a_i \leq x_i \leq b_i, i = 1, \ldots, n\}$, when some physical bounds on the object are imposed.

Bibliography

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover Publications, New York, 1972.

[2] R. Acar and C. R. Vogel. Analysis of bounded variation penalty methods for ill-posed problems. *Inverse Problems*, 10(6):1217–1229, June 1994.

[3] H. Akaike. On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method. *Annals of the Institute of Statistical Mathematics*, 11:1–16, 1959.

[4] R. D. Allen, G. B. David, and G. Nomarski. The Zeiss-Nomarski differential interference equipment for transmitted-light microscopy. *Zeitschrift fur wissenschaftliche Mikroskopie und mikroskopische Technik*, 69(4):193–221, 1969.

[5] M. S. C. Almeida and L. B. Almeida. Blind and semi-blind deblurring of natural images. *IEEE Transactions on Image Processing*, 19(1):36–52, Jan. 2010.

[6] C. Arcidiacono, E. Diolaiti, M. Tordi, R. Ragazzoni, J. Farinato, E. Vernet, and E. Marchetti. Layer-Oriented Simulation Tool. *Applied Optics*, 43(22):4288–4302, Aug. 2004.

[7] H. Attouch and J. Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming*, 116(1–2):5–16, Jan. 2009.

[8] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the Kurdyka-łojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, May 2010.

[9] H. Attouch, J. Bolte, and B. F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods. *Mathematical Programming*, 137(1–2):91–129, Feb. 2013.

[10] V. P. Bailey, P. M. Hinz, A. T. Puglisi, S. Esposito, V. Vaitheeswaran, A. J. Skemer, D. Defrère, A. Vaz, and J. M. Leisenring. Large binocular telescope interferometer adaptive optics: on-sky performance and lessons learned. In E. Marchetti, L. M. Close, and J.-P. Véran, editors, *Adaptive Optics Systems IV*, volume 9148 of *Proceedings of SPIE*, 2014.

[11] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.

[12] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces.* CMS Books on Mathematics. Springer, 2011.

[13] L. Bautista, S. Rebegoldi, L. Blanc-Féraud, M. Prato, L. Zanni, and A. Plata. Phase estimation in differential-interference-contrast (DIC) microscopy. In *IEEE Proceedings of the 13th International Symposium on Biomedical Imaging (ISBI)*, pages 136–139, 2016.

[14] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Science*, 2(1):183–202, 2009.

[15] A. Beck and M. Teboulle. Gradient-based algorithms with applications to signal recovery problems. In D. Palomar and Y. Eldar, editors, *Convex Optimization in Signal Processing and Communications*, page 42–88. Cambribge University Press, 2010.

[16] F. Benvenuto, R. Zanella, L. Zanni, and M. Bertero. Nonnegative least-squares image deblurring: improved gradient projection approaches. *Inverse Problems*, 26(2), Feb. 2010.

[17] M. Bertero and P. Boccacci. *Introduction to inverse problems in imaging.* Institute of Physics Publishing, Bristol, 1998.

[18] M. Bertero, P. Boccacci, G. Desiderà, and G. Vicidomini. Image deblurring with Poisson data: from cells to galaxies. *Inverse Problems*, 25(12), Dec. 2009.

[19] M. Bertero, P. Boccacci, G. Talenti, R. Zanella, and L. Zanni. A discrepancy principle for Poisson data. *Inverse Problems*, 26(10), Oct. 2010.

[20] M. Bertero, H. Lantéri, and L. Zanni. Iterative image reconstruction: a point of view. In Y. Censor, M. Jiang, and A. K. Louis, editors, *Mathematical Methods in Biomedical Imaging and Intensity-Modulated Radiation Therapy (IMRT)*, pages 37–63. Birkhauser-Verlag, Pisa, Italy, 2008.

[21] D. Bertsekas. On the Goldstein-Levitin-Polyak gradient projection method. *IEEE Transactions on Automatic Control*, 21:174–184, 1976.

[22] D. Bertsekas. *Nonlinear programming.* Athena Scientific, Belmont, 1999.

[23] J. C. Bezdek and R. J. Hathaway. Convergence of alternating optimization. *Neural, Parallel and Scientific Computing*, 11(4):351–368, 2003.

[24] E. G. Birgin, J. M. Martinez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10:1196–1211, 2000.

[25] E. G. Birgin, J. M. Martinez, and M. Raydan. Inexact spectral projected gradient methods on convex sets. *IMA Journal of Numerical Analysis*, 23(4):539–559, Oct. 2003.

[26] J. Bochnak, M. Coste, and M.-F. Roy. *Real Algebraic Geometry.* Springer-Verlag, Berlin, 1998.

[27] J. Bolte, A. Daniilidis, and A. Lewis. The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17:1205–1223, 2007.

[28] J. Bolte, A. Daniilidis, A. Lewis, and M. Shiota. Clarke subgradients of stratifiable functions. *SIAM Journal on Optimization*, 18:556–572, 2007.

[29] J. Bolte, A. Daniilidis, O. Ley, and L. Mazet. Characterizations of Łojasiewicz inequalities: Subgradient flows, talweg, convexity. *Transactions of the American Mathematical Society*, 362(6), 2010.

[30] S. Bonettini. Inexact block coordinate descent methods with application to the nonnegative matrix factorization. *IMA Journal of Numerical Analysis*, 31(4):1431–1452, Oct. 2011.

[31] S. Bonettini, A. Cornelio, and M. Prato. A new semiblind deconvolution approach for Fourier-based image restoration: an application in astronomy. *SIAM Journal on Imaging Science*, 6(3):1736–1757, 2013.

[32] S. Bonettini, I. Loris, F. Porta, and M. Prato. Variable metric inexact line–search based methods for nonsmooth optimization. *SIAM Journal on Optimization*, 26(2):891–921, 2016.

[33] S. Bonettini, I. Loris, F. Porta, M. Prato, and S. Rebegoldi. On the convergence of a line-search based proximal-gradient method for nonconvex optimization. *Inverse Problems*, 33:055005, 2017.

[34] S. Bonettini and M. Prato. Nonnegative image reconstruction from sparse Fourier data: a new deconvolution algorithm. *Inverse Problems*, 26(9), Sept. 2010.

[35] S. Bonettini and M. Prato. New convergence results for the scaled gradient projection method. *Inverse Problems*, 31(9):095008, Sept. 2015.

[36] S. Bonettini, M. Prato, and S. Rebegoldi. A cyclic block coordinate descent method with generalized gradient projections. *Applied Mathematics and Computation*, 286:288–300, 2016.

[37] S. Bonettini, R. Zanella, and L. Zanni. A scaled gradient projection method for constrained image deblurring. *Inverse Problems*, 25(1), Jan. 2009.

[38] P. Brianzi, F. Di Benedetto, and C. Estatico. Preconditioned iterative regularization in Banach spaces. *Computational Optimization and Applications*, 54(2):263–282, Mar. 2013.

[39] A. Cassioli, D. Di Lorenzo, and M. Sciandrone. On the convergence of inexact block coordinate descent methods for constrained optimization. *European Journal of Operational Research*, 231(2):274–281, Dec. 2013.

[40] A. Cauchy. Méthode générale pour la résolution des systèmes d'equations simultanées. *Comptes Rendus de l'Académie de Sciences Paris*, 25:536–538, 1847.

[41] A. Chambolle, R. A. DeVore, N. Y. Lee, and B. J. Lucier. Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Transactions on Image Processing*, 7:319–335, 1998.

[42] A. Chambolle and C. Dossal. On the convergence of the iterates of "Fast Iterative Shrinkage/Thresholding Algorithm". *Journal of Optimization Theory and Applications*, 166(3):968–982, Sept. 2015.

[43] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing*, 6(2):298–311, Feb. 1997.

[44] Y. J. Chen, T. Pock, R. Ranftl, and H. Bischof. Matlab code and supplementary material for loss specific training of MRF models, 2013. http://gpu4vision.icg.tugraz.at/index.php?content=downloads.php.

[45] Y. J. Chen, T. Pock, R. Ranftl, and H. Bischof. Revisiting loss-specific training of filter-based MRFs for image restoration. In J. Weickert, M. Hein, and B. Schiele, editors, *Pattern Recognition*, pages 271–281. Springer-Verlag, Berlin, 2013.

[46] E. Chouzenoux and J.-C. Pesquet. A stochastic majorize-minimize subspace algorithm for online penalized least squares estimation. *ArXiv e-prints*, page 1512.08722, 2016.

[47] E. Chouzenoux, J.-C. Pesquet, and A. Repetti. Variable metric forward–backward algorithm for minimizing the sum of a differentiable function and a convex function. *Journal of Optimization Theory and Applications*, 162(1):107–132, July 2014.

[48] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, editors, *Fixed-point algorithms for inverse problems in science and engineering*, Springer Optimization and Its Applications, pages 185–212. Springer, New York, NY, 2011.

[49] P. L. Combettes and B. C. Vũ. Variable metric quasi-Féjer monotonicity. *Nonlinear Analysis: Theory, Methods & Applications*, 78:17–31, Feb. 2013.

[50] P. L. Combettes and B. C. Vũ. Variable metric forward-backward splitting with applications to monotone inclusions in duality. *Optimization*, 63(9):17–31, Sept. 2014.

[51] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4(4):1168–1200, 2005.

[52] A. Cornelio, F. Porta, and M. Prato. A convergent least-squares regularized blind deconvolution approach. *Applied Mathematics and Computation*, 259(12):173–186, May 2015.

[53] Y. H. Dai. Alternate step gradient method. *Optimization*, 52:395–415, 2003.

[54] Y. H. Dai and R. Fletcher. Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. *Numerische Mathematik*, 100:21–47, 2005.

[55] Y. H. Dai and R. Fletcher. New algorithms for singly linearly constrained quadratic programming problems subject to lower and upper bounds. *Mathematical Programming*, 106(3):403–421, May 2006.

[56] Y. H. Dai and L. Z. Liao. R-linear convergence of the Barzilai and Borwein gradient method. *IMA Journal of Numerical Analysis*, 22:1–10, 2002.

[57] Y. H. Dai and Y. X. Yuan. Alternate minimization gradient method. *IMA Journal of Numerical Analysis*, 23(3):377–393, July 2003.

[58] I. Daubechies, M. Defrise, and C. D. Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, (57):1413–1457, 2004.

[59] G. Desiderà and M. Carbillet. Strehl-constrained iterative blind deconvolution for post-adaptive-optics data. *Astronomy & Astrophysics*, 507(3):1759–1762, Dec. 2009.

[60] S. Esposito, A. Riccardi, L. Fini, A. T. Puglisi, E. Pinna, M. Xompero, R. Briguglio, F. Quirós-Pacheco, P. Stefanini, C. J. Guerra, L. Busoni, A. Tozzi, F. Pieralli, G. Agapito, G. Brusa-Zappellini, R. Demers, J. Brynnel, C. Arcidiacono, and P. Salinari. First light AO (FLAO) system for LBT: final integration, acceptance, test in Europe, and preliminary on-sky commissioning results. In L. B. Ellerbroek, M. Hart, N. Hubin, and P. L.

Wizinowich, editors, *Adaptive Optics Systems II*, volume 7736 of *Proceedings of SPIE*, 2010.

[61] S. Esposito, A. Riccardi, E. Pinna, A. T. Puglisi, F. Quirós-Pacheco, C. Arcidiacono, M. Xompero, R. Briguglio, L. Busoni, L. Fini, J. Argomedo, A. Gherardi, G. Agapito, G. Brusa, D. L. Miller, J. C. Guerra Ramon, K. Boutsia, and P. Stefanini. Natural guide star adaptive optics systems at LBT: FLAO commissioning and science operations status. In L. B. Ellerbroek, E. Marchetti, and J.-P. Véran, editors, *Adaptive Optics Systems III*, volume 8447 of *Proceedings of SPIE*, 2012.

[62] R. Fletcher. Low storage methods for unconstrained optimization. *Lectures in Applied Mathematics*, 26:165–179, 1990.

[63] R. Fletcher. *Practical methods of optimization.* John Wiley and Sons, New York, 2nd edition, 2000.

[64] R. Fletcher. On the Barzilai-Borwein method. *Optimization and Control with Applications*, 96:235–256, 2005.

[65] R. Fletcher. A limited memory steepest descent method. *Mathematical Programming*, 135(1–2):413–436, Oct. 2012.

[66] P. Frankel, G. Garrigos, and J. Peypouquet. Splitting methods with variable metric for Kurdyka–Łojasiewicz functions and general convergence rates. *Journal of Optimization Theory and Applications*, 165(3):874–900, June 2015.

[67] G. Frassoldati, G. Zanghirati, and L. Zanni. New adaptive stepsize selections in gradient methods. *Journal of Industrial and Management Optimization*, 4(2):299–312, 2008.

[68] A. Friedlander, J. Martinez, B. Molina, and M. Raydan. Gradient method with retards and generalizations. *SIAM Journal on Numerical Analysis*, 36:275–289, 1999.

[69] I. Galic, J. Weickert, M. Welk, A. Bruhn, A. G. Belyaev, and H.-P. Seidel. Image compression with anisotropic diffusion. *Journal of Mathematical Imaging and Vision*, 31(2–3):255–269, 2008.

[70] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, Nov. 1984.

[71] J. C. Gilbert and J. Nocedal. Global convergence properties of conjugate gradient methods for optimization. *SIAM Journal on Optimization*, 2(1):21–42, 1992.

[72] W. Glunt, T. Hayden, and M. Raydan. Molecular conformations from distance matrices. *Journal of Computational Chemistry*, 14:114–120, 1993.

[73] G. H. Golub and C. F. Van Loan. *Matrix Computations.* John Hopkins University Press, Baltimore, 3rd edition, 1996.

[74] J. W. Goodman. *Statistical Optics.* Wiley, New York, 1984.

[75] L. Grippo, F. Lampariello, and S. Lucidi. A nonmonotone line search technique for Newton's method. *SIAM Journal on Numerical Analysis*, 23(4):707–716, 1986.

[76] L. Grippo and M. Sciandrone. Globally convergent block-coordinate techniques for unconstrained optimization. *Optimization Methods and Software*, 10(4):587–637, 1999.

[77] L. Grippo and M. Sciandrone. On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. *Operations Research Letters*, 26(3):127–136, Apr. 2000.

[78] J. Hadamard. *Lectures on Cauchy's problem in linear partial differential equations.* Yale University Press, New Haven, 1923.

[79] S. B. Hadj, L. Blanc-Féraud, and G. Aubert. Space variant blind image restoration. *SIAM Journal on Imaging Science*, 7(4):2196–2225, 2014.

[80] P. C. Hansen. *Rank-deficient and discrete ill-posed problems.* SIAM, Philadelphia, 1997.

[81] P. C. Hansen, J. G. Nagy, and D. P. O'Leary. *Deblurring Images: Matrices, Spectra and Filtering.* SIAM, Philadelphia, 2006.

[82] Z. T. Harmany, R. F. Marcia, and R. M. Willett. This is spiral-tap: sparse Poisson intensity reconstruction algorithms–theory and practice. *IEEE Transactions on Image Processing*, 3(21):1084–1096, Mar. 2012.

[83] T. Herbst, R. Ragazzoni, D. Andersen, H. Boehnhardt, P. Bizenberger, A. Eckart, W. Gaessler, H.-W. Rix, R.-R. Rohloff, P. Salinari, R. Soci, C. Straubmeier, and W. Xu. LINC-NIRVANA: a Fizeau beam combiner for the large binocular telescope. In W. A. Traub, editor, *Interferometry for Optical Astronomy II*, volume 4838 of *Proceedings of SPIE*, pages 456–465, 2003.

[84] J. B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms. II.* Springer–Verlag, Berlin, 1993.

[85] L. Hoeltgen, S. Setzer, and J. Weickert. An optimal control approach to find sparse data for Laplace interpolation. In A. Heyden, F. Kahl, C. Olsson, M. Oskarsson, and X.-C. Tai, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 151–164. Springer–Verlag, Berlin, 2013.

[86] A. N. Iusem. On the convergence properties of the projected gradient method for convex optimization. *Computational and Applied Mathematics*, 22(1):37–52, 2003.

[87] D. Kim, S. Sra, and I. S. Dhillon. Fast Newton-type methods for the least squares nonnegative matrix approximation problem. In *SIAM International Conference on Data Mining*, pages 343–354, 2007.

[88] S. Krantz and H. R. Parks. *A primer of Real Analytic Functions*. Birkhäuser, 2002.

[89] H. Lantéri, M. Roche, and C. Aime. Penalized maximum likelihood image restoration with positivity constraints: multiplicative algorithms. *Inverse Problems*, 18(5):1397–1419, Oct. 2002.

[90] H. Lantéri, M. Roche, O. Cuevas, and C. Aime. A general method to devise maximum likelihood signal restoration multiplicative algorithms with non-negativity constraints. *Signal Processing*, 81(5):945–974, May 2001.

[91] L. Lecharlier and C. De Mol. Regularized blind deconvolution with Poisson data. In L. Blanc-Féraud and P.-Y. Joubert, editors, *3rd International Workshop on New Computational Methods for Inverse Problems*, volume 464 of *Journal of Physics: Conference Series*, page 012003, 2013.

[92] C. J. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, 19(10):2756–2779, Oct. 2007.

[93] S. Łojasiewicz. Une propriété topologique des sous-ensembles analytiques réels. In *Les Équations aux Dérivées Partielles*, pages 87–89. Éditions du Centre National de la Recherche Scientifique, Paris, 1963.

[94] I. Loris, M. Bertero, C. De Mol, R. Zanella, and L. Zanni. Accelerating gradient projection methods for ℓ_1-constrained signal recovery by steplength selection rules. *Applied and Computational Harmonic Analysis*, 27(2):247–254, Sept. 2009.

[95] L. B. Lucy. An iterative technique for the rectification of observed distributions. *Astronomical Journal*, 79(6):745–754, June 1974.

[96] D. Luenberger. *Introduction to Linear and Nonlinear Programming*. Addison Wesley, 2nd edition, 1984.

[97] V. N. Mahajan. Strehl ratio for primary aberrations in terms of their aberration variance. *Journal of the Optical Society of America*, 73(6):860–861, June 1983.

[98] S. B. Mehta and C. J. R. Sheppard. Partially coherent image formation in differential interference contrast (DIC) microscope. *Optics Express*, 16(24):19462–19479, Nov. 2008.

[99] J.-J. Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *Comptes Rendus de l'Académie des Sciences (Paris) Série A*, 255:2897–2899, 1962.

[100] D. B. Murphy. *Fundamentals of light microscopy and electronic imaging.* Wiley-Liss, New York, 2001.

[101] A. Nemirovski. Prox-method with rate of convergence $o(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex–concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

[102] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Doklady Akademii Nauk SSSR*, 27:372–376, 1983.

[103] J. Nocedal, A. Sartenaer, and C. Zhu. On the behavior of the gradient norm in the steepest descent method. *Computational Optimization and Applications*, 22(1):5–35, 2002.

[104] J. Nocedal and S. J. Wright. *Numerical optimization.* Springer, New York, 2nd edition, 2006.

[105] P. Ochs, Y. Chen, T. Brox, and T. Pock. iPiano: Inertial proximal algorithm for nonconvex optimization. *SIAM Journal on Imaging Science*, 7(2):1388–1419, 2014.

[106] B. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4:1–17, 1964.

[107] F. Porta, M. Prato, and L. Zanni. A new steplength selection for scaled gradient methods with application to image deblurring. *Journal of Scientific Computing*, 65(3):895–919, Dec. 2015.

[108] F. Porta, R. Zanella, G. Zanghirati, and L. Zanni. Limited-memory scaled gradient projection methods for real-time image deconvolution in microscopy. *Communications in Nonlinear Science and Numerical Simulation*, 21(1–3):112–127, Apr. 2015.

[109] M. J. D. Powell. On search directions for minimization algorithms. *Mathematical Programming*, 4(1):193–201, Dec. 1973.

[110] M. Prato, S. Bonettini, A. La Camera, and S. Rebegoldi. Alternating minimization for Poisson blind deconvolution in astronomy. In *Proceedings of the Inverse Problems from Theory to Applications Conference (IPTA 2014)*, pages 148–152, 2014.

[111] M. Prato, S. Bonettini, I. Loris, F. Porta, and S. Rebegoldi. On the constrained minimization of smooth Kurdyka–Łojasiewicz functions with the scaled gradient projection method. In *Journal of Physics: Conference Series*, volume 756, page 012004, 2016.

[112] M. Prato, R. Cavicchioli, L. Zanni, P. Boccacci, and M. Bertero. Efficient deconvolution methods for astronomical imaging: algorithms and IDL-GPU codes. *Astronomy & Astrophysics*, 539:A133, Mar. 2012.

[113] M. Prato, A. La Camera, S. Bonettini, and M. Bertero. A convergent blind deconvolution method for post-adaptive-optics astronomical imaging. *Inverse Problems*, 29(6), June 2013.

[114] M. Prato, A. La Camera, S. Bonettini, S. Rebegoldi, M. Bertero, and P. Boccacci. A blind deconvolution method for ground based telescopes and fizeau interferometers. *New Astronomy*, 40:1–13, Oct. 2015.

[115] C. Preza. Rotational-diversity phase estimation from differential interference contrast microscopy images. *Journal of the Optical Society of America A*, 17(3):415–424, 2000.

[116] C. Preza, S. V. King, and C. J. Cogswell. Algorithms for extracting true phase from rotationally-diverse and phase-shifted DIC images. In J.-A. Conchello, C. J. Cogswell, and T. Wilson, editors, *Three-Dimensional and Multidimensional Microscopy: Image Acquisition and Processing XIII*, volume 6090 of *Proceedings of SPIE*, 2006.

[117] C. Preza, D. L. Snyder, and J.-A. Conchello. Image reconstruction for three-dimensional transmitted-light DIC microscopy. In J.-A. Conchello, C. J. Cogswell, and T. Wilson, editors, *Three-Dimensional Microscopy: Image Acquisition and Processing IV*, volume 2984 of *Proceedings of SPIE*, 1997.

[118] C. Preza, D. L. Snyder, and J.-A. Conchello. Theoretical development and experimental evaluation of imaging models for differential interference contrast microscopy. *Journal of the Optical Society of America A*, 16(9):2185–2199, 1999.

[119] B. A. Probes. Product description APCS-0099. http://www.brukerafmprobes.com/a-3472-apcs-0099.aspx, July 2016.

[120] M. Raydan. On the Barzilai and Borwein choice of steplength for the gradient method. *IMA Journal of Numerical Analysis*, 13:321–326, 1993.

[121] M. Raydan. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM Journal on Optimization*, 7:26–33, 1997.

[122] M. Raydan and B. Svaiter. Relaxed steepest descent and Cauchy-Barzilai-Borwein method. *Computational Optimization and Applications*, 21:155–167, 2002.

[123] M. Razaviyayn, M. Hong, and Z.-Q. Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013.

[124] S. Rebegoldi. Matlab toolbox for DIC microscopy, 2017. http://www.oasis.unimore.it/site/home/software.html.

[125] S. Rebegoldi, L. Bautista, L. Blanc-Féraud, M. Prato, L. Zanni, and A. Plata. TV-regularized phase reconstruction in differential-interference-contrast (DIC) microscopy. In *AIP Conference Proceedings*, volume 1776, page 090043, 2016.

[126] S. Rebegoldi, S. Bonettini, and M. Prato. Application of cyclic block generalized gradient projection methods to Poisson blind deconvolution. In *Proceedings of the 23rd European Signal Processing Conference*, pages 225–229, 2015.

[127] A. Repetti. Toolbox Matlab de restauration d'images par l'algorithme VMFB, 2013. http://www-syscom.univ-mlv.fr/~chouzeno/Logiciel.html.

[128] W. H. Richardson. Bayesian based iterative method of image restoration. *Journal of the Optical Society of America*, 62(1):55–59, Jan. 1972.

[129] R. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

[130] R. T. Rockafellar. *Convex analysis.* Princeton University Press, Princeton, NJ, 1970.

[131] R. T. Rockafellar, R. J.-B. Wets, and M. Wets. *Variational Analysis*, volume 317 of *Grundlehren der Mathematischen Wissenschaften.* Springer, Berlin, 1998.

[132] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1–4):259–268, 1992.

[133] S. Salzo and S. Villa. Inexact and accelerated proximal point algorithms. *Journal of Convex Analysis*, 19(4):1167–1192, Dec. 2012.

[134] F. Sciacchitano, Y. Dong, and T. Zeng. Variational approach for restoring blurred images with Cauchy noise. *SIAM Journal on Imaging Science*, 8(3):1894–1922, 2015.

[135] L. A. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Transactions on Medical Imaging*, 1(2):113–122, Oct. 1982.

[136] D. L. Snyder, C. W. Helstrom, A. D. Lanterman, M. Faisal, and R. L. White. Compensation for readout noise in CCD images. *Journal of the Optical Society of America A*, 12(2):272–283, Feb. 1995.

[137] A. Staglianò, P. Boccacci, and M. Bertero. Analysis of an approximate model for poisson data reconstruction and a related discrepancy principle. *Inverse Problems*, 27(12), Dec. 2011.

[138] N. A. Tikhonov and V. Y. Arsenin. *Solution of Ill Posed Problems.* Wiley, New York, 1977.

[139] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109:475–494, 2001.

[140] P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1–2):387–423, Mar. 2009.

[141] C. R. Vogel. *Computational methods for inverse problems.* SIAM, Philadelphia, 2002.

[142] J. C. Wilson, P. M. Hinz, M. F. Skrutskie, T. Jones, E. Solheid, J. Leisenring, P. Garnavich, M. Kenwhorty, M. J. Nelson, and C. E. Woodward. LMIRcam: an L/M-band imager for the LBT combined focus. In M. Schoeller, W. C. Dauchi, and F. Delplancke, editors, *Optical and Infrared Interferometry*, volume 7013 of *Proceedings of SPIE*, 2008.

[143] A. Zalinescu. *Convex analysis in general vector spaces.* World Scientific Publishing Co. Inc., River Edge, NJ, 2002.

[144] R. Zanella, P. Boccacci, L. Zanni, and M. Bertero. Efficient gradient projection methods for edge-preserving removal of Poisson noise. *Inverse Problems*, 25(4), Apr. 2009.

[145] B. Zhou, L. Gao, and Y. H. Dai. Gradient methods with adaptive step-sizes. *Computational Optimization and Applications*, 35(1):69–86, Sept. 2006.