Università degli Studi di Ferrara

Dottorato di Ricerca in Matematica

Ciclo XXX

Coordinatore Prof. Massimiliano Mella

Variable metric first–order methods for applications in biomedical imaging

Settore Scientifico Disciplinare MAT/08

Dottoranda

Dott.ssa Vanna Lisa Coli

Tutor

Prof. Luca Zanni

Anni 2014/2017

Contents

Introduction

Optimization algorithms are extensively employed in the resolution of problems that arise in a wide variety of applied scientific domains.

Many signal reconstruction problems in seismology, biomedicine, acoustics, microscopy and astronomy can be formulated as inverse problems and tackled by optimizing an appropriate function dependent on some properties of the unknown signal. In many instances, additional information can be employed by introducing constraints on the signal to be reconstructed. Most of the applied inverse problems are large–scale ones and their solution cannot be computed in a very accurate way by direct methods, thus they are addressed by iterative procedures whose computing effort is governed by a given tolerance. Some applications, for example in the biomedical domain, require as a further challenge the determination of solutions in almost real–time. As a consequence, the definition of numerical optimization methods whose accuracy can be considered acceptable in a short time has become a crucial issue.

Because of their simple implementation and relatively light computational burden, first–order methods are often the choice when accuracy requirements on the solution are not severe. Despite their popularity, one of the major drawback of this class of methods is the slow convergence rate towards the optimal solution; thus, computational techniques able to speed up the performances of such methods are a stimulating research topic.

Biomedical imaging has received growing attention in the clinical field thanks to the recent advances in technology and signal processing, so that nowadays it is considered as an own domain of research.

The development of reconstruction techniques in this field strongly involves mathematical tools, such as modeling of imaging devices or noise handling procedures. Moreover, the amount of acquired data is typically large, so that computational load and memory requirements have to be carefully organized. Furthermore, one of the practical issues in some special biomedical imaging systems (for example, those related to Computed Tomography) is the limitation on measurement time, due to the necessity of keeping dose prescriptions low in order to preserve the wellness of the patient; thus, the reconstruction of high resolution images must deal with this reduction of the information amount.

For all these reasons, first–order numerical methods, based on low storage requirements and very simple iterations, appear as possible candidates to encounter practical issues arising in this image reconstruction context.

The research activity presented in this thesis has dealt with the analysis of acceleration techniques for first–order methods in nonlinear constrained optimization and their impact in signal reconstruction problems arising in some biomedical domains. The work mainly concerned the study of recent strategies that involve steplengths selection rules and variable metrics induced by scaling matrices; these techniques make use of tools that are already furnished by the methods themselves, thus they preserve the agility of the first–order algorithms without increasing their computational costs. Further work was devoted to the study of regularization methods for inverse problems, with the aim of extending the aforementioned acceleration techniques to a wider class of problems.

The thesis is organised in four chapters and could be ideally splitted in two main parts: a first part where a theoretical framework for first–order optimization methods for inverse problems is designed and a second one where the algorithms are put into action on applicative problems.

In Chapter 1, we give the generalities of signal restoration problems, providing some details on the signal formation process and on the noise arising during the data acquisition. Moreover, we resume the basics of the statistical framework underlying the solution of inverse problems, focusing particularly on the maximum likelihood approach and on the regularization functionals which will be employed in the considered biomedical imaging applications.

In Chapter 2, we give a survey of forward–backward methods for constrained optimization problems, recalling some state–of–the–art algorithm and other methods developed in recent years. A more accurate analysis is devoted to gradient methods and possible strategies able to speed up the convergence rate are discussed; in particular, recent techniques based on the introduction of a variable metric and on the choice of the steplength parameter are detailed for a classical gradient projection method and for a gradient projection method with extrapolation step.

In Chapter 3, the first application framework for the methods proposed in Chapter 2 is presented. The problem concerns the reconstruction of fibre orientation distribution on the cerebral white matter from diffusion Magnetic Resonance Imaging data. A recent paper presented a Spherical Deconvolution–based approach that aims to solve the given problem by means of a constrained ℓ_0 minimization formulation, which envisages to solve a sequence of Least Squares constrained problems subjected to nonnegativity and sparsity constraints; the variable metric algorithms described in Chapter 2 (namely the Scaled Gradient Projection (SGP) and the Scaled Gradient Projection with Extrapolation (Scaled GP_Ex)) are employed to tackle the described problems and their performances are estimated in comparison with a state–of–the–art method.

In Chapter 4, several acceleration strategies designed for the SGP algorithm are applied to a problem of reconstruction of 3D X–ray tomographic images from limited data. The problem is formulated as the nonnegatively constrained minimization of an objective function expressed by the sum of a fit–to–data term and a smoothed Total Variation function. The choice of the data fitting function is strictly related to the noise that affects real Computed Tomography systems; thus different functionals were considered in order to evaluate the behaviour of the methods on realistic scenarios. The numerical results obtained on a 3D phantom are presented and a state–of–the–art method for Computed Tomography problems is considered to perform comparisons.

List of Algorithms

Notations

- $\mathbb{R}_{\geq 0} = \{\boldsymbol{x} \in \mathbb{R} : \ \boldsymbol{x} \geq \mathbf{0}\}$ and $\mathbb{R}_{>0} = \{\boldsymbol{x} \in \mathbb{R} : \ \boldsymbol{x} > \mathbf{0}\}$ 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 $\mathbf{0}$ denote a vector with all entries equal to 1 and 0, respectively.
- If $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, then $\boldsymbol{x}^T\boldsymbol{y} = \sum_{i=1}^n x_i y_i$ denotes the scalar product.
- If $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, then $\frac{\boldsymbol{x}}{\boldsymbol{y}}$ and $\boldsymbol{x} \cdot \boldsymbol{y}$ denote the component-wise division and product, respectively.
- If $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{x} \geq \mathbf{0} \Leftrightarrow x_i \geq 0, \ i = 1, \ldots, n$. 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.
- $\|\cdot\|$ denotes the Euclidean norm: $\|\boldsymbol{x}\| = \|\boldsymbol{x}\|_2 = \sqrt{\boldsymbol{x}^T\boldsymbol{x}}$.
- $\|\cdot\|_D$ denotes the norm induced by a symmetric positive definite matrix $D \in \mathbb{R}^{n \times n}$: $\|\boldsymbol{x}\|_D = \sqrt{\boldsymbol{x}^T D\boldsymbol{x}}$.
- 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 $A, B \in \mathbb{R}^{n \times n}$ symmetric positive semidefinite matrices, the notation $A \succeq B$ indicates that $A - B$ is positive semidefinite.

Chapter 1

General techniques for signal and image reconstruction

Signal reconstruction is a major problem in several domains of applied science, such as, for instance, seismology, acoustic, astronomy and medical imaging. In many practical cases, it is expressed as a linear inverse and ill–posed problem; as it is now frequent to process data of the order of several billions, restoration problems are usually large–scale ones. For all these reasons, efficient numerical methods need to be applied and, due to the dimension of the unknown signal, iterative methods represent the most exploited choice, apart from some special instances.

In this chapter we will give a synopsis of literature results concerning imaging restoration problems, but the ideas and the methods presented can be extended to more general types of signals. In Section 1.1 we recall the mathematical framework for modelling the reconstruction problem, followed in Section 1.2 by the description of the statistical approaches that allow to tackle it. In Section 1.3 we report some regularization approaches devoted to limiting noise effects on acquired data and in Section 1.4 we list some constraints commonly employed in reconstruction problems.

1.1 Data restoration

Image reconstruction is a well–known example of ill–posed problem, thus, it requires an accurate mathematical modeling in order to obtain a proper formulation of the problem. A general imaging system can be modeled in two parts, as follows:

- a physical apparatus (composed by material components such as sources, mirrors, lenses etc.), which is capable to convert the radiation (photons, X–rays, microwaves, ultrasounds etc.) emitted by the *object* to be imaged into a detectable radiation containing information about the properties of the object;

- a detector which provides measured values of the incoming radiation, thus introducing sampling and noise.

Image reconstruction techniques are based on an *image formation model*, which describes the propagation of the radiation used in the imaging process. The optical image formation is frequently modelled by the following continuous linear model [10]

$$\bar{y}(s) = \int \mathcal{H}(s, s')x(s')ds' \tag{1.1}$$

where $x(s)$ denotes a function of the space variables describing the unknown *object*, $\bar{y}(s)$ represents the acquired *noise–free image* and $\mathcal{H}(s, s')$ is the *Point Spread Function (PSF).* The term comes from the assumption that $\mathcal{H}(\cdot, s')$ is the image of a point source located at the point s'. In fact, if $\delta(\cdot)$ indicates the Delta distribution and the object is given by $u(s'') = \delta(s'' - s')$, then from (1.1) one can obtain $\bar{y}(s) = \mathcal{H}(s, s')$. The effect of the PSF is called *blurring* and $\bar{y}$ is the blurred image.

In many 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 (1.1) reduces to

$$\bar{y}(s) = \int \mathcal{H}(s - s')x(s')ds' = (\mathcal{H} \otimes x)(s) \tag{1.2}$$

where $\otimes$ denotes the convolution product.

Moreover, a discrete version of model (1.2) is required when images are treated as digital signals. In this case, the unknown object and the PSF will be two vectors $\boldsymbol{x}, \boldsymbol{h} \in \mathbb{R}^n$, and the convolution product can be seen as the matrix–vector product $H\boldsymbol{x}$, where $H \in \mathbb{R}^{m \times n}$ is the convolution matrix obtained by imposing some specific boundary conditions on the discretised PSF $\boldsymbol{h}$ [66]. Finally, if we take into account the presence of noise due to recording process and we consider a nonnegative constant *background term* b into the model, we can write

$$\boldsymbol{y} = H\boldsymbol{x} + b\boldsymbol{e} + \boldsymbol{v} \tag{1.3}$$

where $\boldsymbol{y} \in \mathbb{R}^m$ is the blurred and noisy image and $\boldsymbol{v} \in \mathbb{R}^m$ represents the additive noise contribution.

The following standard assumptions can be made on the convolution matrix H:

$$H_{i,j} \geq 0, \ \forall\, i, j, \quad H^T\boldsymbol{e} = \boldsymbol{e}, \quad H\boldsymbol{e} > 0.$$

In other words, we assume that each row or column contains at least one nonzero component. Furthermore, if *periodic* boundary conditions are employed in model (1.3), i.e. if the two–dimensional PSF $\boldsymbol{h} = \{h_{i,j}\}_{i=1,\ldots,m}^{j=1,\ldots,n}$ is such that

$$h_{m+1,j} = h_{1,j}, \quad h_{i,m+1} = h_{i,1}, \quad \forall\, i, j$$

then H is block circulant with circulant blocks and the matrix–vector products $H\boldsymbol{x}$ and $H^T\boldsymbol{x}$ can be efficiently computed by making use of the Discrete Fourier Transform (DFT) and its inverse (IDFT) [10, 66]. In fact, from the convolution theorem, we have

$$\begin{aligned} H\boldsymbol{x} &= \mathrm{IDFT}\left(\mathrm{DFT}(\boldsymbol{h}) \cdot \mathrm{DFT}(\boldsymbol{x})\right) \\ H^T\boldsymbol{x} &= \mathrm{IDFT}\left(\overline{\mathrm{DFT}(\boldsymbol{h})} \cdot \mathrm{DFT}(\boldsymbol{x})\right), \end{aligned}$$

where $\overline{\alpha}$ denotes the complex conjugate of $\alpha \in \mathbb{C}$. The above matrix–vector products may be performed with a $\mathcal{O}(mn\log(mn))$ complexity by means of the Fast Fourier Transform (FFT) algorithm implementation [66, 115]. This efficient computation is guaranteed also with other boundary conditions, such as zero or reflexive conditions, which involve the use of other discrete transforms apart from the DFT [66]. For these reasons, gradient–based reconstruction algorithms are computationally sustainable, as gradient evaluations typically require matrix–vector products.

The noise vector $\boldsymbol{v}$ in model (1.3) can be seen as a realisation of a random variable and, as a consequence, each pixel y_i of the acquired image can be seen as a realisation of a random variable Y_i. If we set $\boldsymbol{Y} = (Y_1, \ldots, Y_m)$, the modelling of the system is then related to the probability density of the multivariate random variable $\boldsymbol{Y}$. This density depends on the object $\boldsymbol{x}$ and therefore we will denote it as $p_{\boldsymbol{Y}}(\boldsymbol{y}; \boldsymbol{x})$. The following statements are typically assumed on Y_i and $\boldsymbol{Y}$ [10, 11]:

- the random variables Y_i are statistically independent, as follows
$$p_{\boldsymbol{Y}}(\boldsymbol{y}; \boldsymbol{x}) = \prod_{i=1}^{m} p_{Y_i}(y_i; \boldsymbol{x});$$
- the expected value of Y_i is given by the i–th pixel of the noise–free image, hence
$$E(\boldsymbol{Y}) = \int \boldsymbol{y}\; p_{\boldsymbol{Y}}(\boldsymbol{y}; \boldsymbol{x})\; d\boldsymbol{y} = H\boldsymbol{x} + b\boldsymbol{e}.$$

In the following we report two common examples of noise modelling.

Example 1.1 (White Gaussian noise)**.** In this first Example the random variable $\boldsymbol{Y}$ is given by

$$\boldsymbol{Y} = H\boldsymbol{x} + b\boldsymbol{e} + \boldsymbol{v}$$

assuming that each component v_i of the noise vector $\boldsymbol{v}$ is a realisation of a random variable with Gaussian distribution of zero mean and standard deviation $\sigma > 0$. In this case, the vector $\boldsymbol{v}$ is a realisation of the multivariate random variable $\boldsymbol{V}$, whose probability density is

$$p_{\boldsymbol{V}}(\boldsymbol{v}) = \frac{1}{(2\pi\sigma^2)^{m/2}} \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{v}\|^2\right).$$

Therefore the statistical model for the detected image is

$$p_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{x}) = \frac{1}{(2\pi\sigma^2)^{m/2}} \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{y} - (H\boldsymbol{x} + b\boldsymbol{e})\|^2\right).$$

Example 1.2 (Poisson noise)**.** This Example can describe the noise affecting counting processes, so that sometimes it is also called *photon noise.* Each Y_i is a Poisson random variable with expected value given by $(H\boldsymbol{x} + b\boldsymbol{e})_i$

$$Y_i \sim \text{Poisson}\{(H\boldsymbol{x} + b\boldsymbol{e})_i)\}$$

As the random variables Y_i are statistically independent, in this case the probability density is a distribution with support the set of nonnegative integers, as each y_i is a nonnegative integer. We have

$$p_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{x}) = \prod_{i=1}^{m} \frac{e^{-(H\boldsymbol{x}+b\boldsymbol{e})_i}(H\boldsymbol{x} + b\boldsymbol{e})_i^{y_i}}{y_i!}.$$

In conclusion, we have a complete model of the process of data formation and acquisition when we know the imaging matrix H, the background b and the probability density $p_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{x})$.

1.2 Statistical formulation of image reconstruction problems

Image restoration is an example of *inverse problem* [10, 63]: if we assume to have a complete model in the sense previously specified and a detected image $\boldsymbol{y}$ (i.e. a realisation of the random variable $\boldsymbol{Y}$), the image reconstruction problem is to recover an estimate of the unknown object $\bar{\boldsymbol{x}}$ corresponding to the image $\boldsymbol{y}$. A naive approach for this problem is to compute

$$\boldsymbol{x} = H^{-1}(\boldsymbol{y} - b\boldsymbol{e})$$

as the solution of the linear system $H\boldsymbol{x} = \boldsymbol{y} - b\boldsymbol{e}$. Nevertheless, this approach is not applicable 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.* Cases are known which employ the direct inversion of the linear model (1.3), for example by means of the *filtered back–projection* algorithm in Computed Tomography, but they are rare exceptions. As informations on statistical properties of the data are available, statistical approaches for this problem arise quite naturally.

1.2.1 Maximum likelihood approach

The first assumption is on the knowledge of the probability density $p_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{x})$, thus it is natural to look for statistical formulations of the image restoration problem. The standard approach is the so–called *maximum likelihood* (ML) estimation [103], in which an estimate of the unknown object $\boldsymbol{x}$ is any $\boldsymbol{x}^*$ that maximizes the probability density of $\boldsymbol{y}$, denominated the *likelihood function* of the problem:

$$\boldsymbol{x}^* = \operatorname*{argmax}_{\boldsymbol{x}\in\mathbb{R}^n} p_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{x}).$$

Equivalenty, one can minimize the negative logarithm of the probability density. Therefore, the ML problem may be reformulated in the following alternative way:

$$\boldsymbol{x}^* = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^n}\; f_0(\boldsymbol{x};\boldsymbol{y}) \equiv -A\ln(p(\boldsymbol{y};\boldsymbol{x})) + B \tag{1.4}$$

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 fit–to–data functionals f_0; we reconsider now the two examples introduced in the previous section.

Example 1.3 (Gaussian noise)**.** If we set set $A = \sigma^2$ and $B = A/(2\pi\sigma^2)^{m/2}$, we have that

$$f_0(\boldsymbol{x};\boldsymbol{y}) = \frac{1}{2}\|H\boldsymbol{x} + b\boldsymbol{e} - \boldsymbol{y}\|^2 \tag{1.5}$$

so that the ML approach leads to the classical *Least Squares* (LS) minimization problem. The functional in (1.5) is convex and is strictly convex if and only if the equation $H\boldsymbol{x} = \boldsymbol{0}$ has only the solution $\boldsymbol{x} = \boldsymbol{0}$. Furthermore, it has always a global minima, i.e. the LS problem has always a solution; nevertheless, in the case of image reconstruction the problem is ill–conditioned, since it is equivalent to the solution of the Euler equation

$$H^T H\boldsymbol{x} = H^T(\boldsymbol{y} - \boldsymbol{x}) \tag{1.6}$$

where the condition number of the matrix H can be very large. In fact, as the matrix H comes from the discretization of an integral operator, the continuous version of this problem is ill–posed, which is a starting point of the Tikhonov regularization theory [50, 112].

Example 1.4 (Poisson noise)**.** If we introduce the Kullback–Leibler (KL) divergence of a vector $\boldsymbol{y}$ from a vector $\boldsymbol{x}$, defined by

$$\mathrm{KL}(\boldsymbol{x};\boldsymbol{y}) = \sum_{i=1}^{m}\left(x_i \ln\left(\frac{x_i}{y_i}\right) + y_i - x_i\right)$$

we can use Stirling's formula $\ln(n!) \approx n\ln(n) - n$ to obtain the following expression for the functional $f_0(\boldsymbol{x};\boldsymbol{y})$

$$\begin{aligned} f_0(\boldsymbol{x};\boldsymbol{y}) &= \mathrm{KL}(\boldsymbol{y}; H\boldsymbol{x} + b\boldsymbol{e}) \qquad (1.7)\\ &= \sum_{i=1}^{m}\left(y_i \ln\left(\frac{y_i}{(H\boldsymbol{x} + b\boldsymbol{e})_i}\right) + (H\boldsymbol{x} + b\boldsymbol{e})_i - y_i\right). \end{aligned}$$

In this case, the domain of f_0 is the nonnegative orthant; moreover, the function is convex and strictly convex if the equation $H\boldsymbol{x} = \boldsymbol{0}$ has only the solution $\boldsymbol{x} = \boldsymbol{0}$ [103], it is nonnegative and locally bounded: thus, it has global minima. Accurate analysis of the properties of this functional can be found in [85, 86]; in particular, an example can be found in [85] where the functional doesn't have a minimum in the classical sense, hence proving the ill–posedness of this minimization problem. Thus, noise is expected to strongly affect the minima of the discrete problem; this is typically confirmed by the effect of the noise which is known as *checkerboard effect*, related to the fact that many components of the minima are zero.

It is worth noticing that, for the considered types of noise, problem (1.4) is still affected by the possible ill–conditioning of the matrix H [10] or by ill–posedness. 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 risky. On the other hand, acceptable (regularized) solutions can be provided by first–order methods with early stopping.

1.2.2 Maximum A Posteriori approach

A comprehensive statistical framework is provided by the *Bayesian approach* [58], in which the unknown object $\boldsymbol{x}$ is also assumed to be a realisation of a multivariate random variable $\boldsymbol{X}$. The probability density of $\boldsymbol{X}$ is the so–called *prior*, which will be denoted by $p_{\boldsymbol{X}}(\boldsymbol{x})$. If the marginal probability $p_{\boldsymbol{Y}}(\boldsymbol{y})$ is introduced, the *Bayes theorem* allows to compute the conditional probability of $\boldsymbol{X}$ with respect to the given value $\boldsymbol{y}$ of $\boldsymbol{Y}$:

$$p_{\boldsymbol{X}}(\boldsymbol{x}|\boldsymbol{y}) = \frac{p_{\boldsymbol{Y}}(\boldsymbol{y}|\boldsymbol{x})p_{\boldsymbol{X}}(\boldsymbol{x})}{p_{\boldsymbol{Y}}(\boldsymbol{y})}.$$

Thus, some properties of the object (such as smoothness, sharp edges etc.) can be incorporated in the *a priori* probability $p_{\boldsymbol{X}}(\boldsymbol{x})$. The most frequently used priors are the Gibbs–type ones:

$$p_{\boldsymbol{X}}(\boldsymbol{x}) = c \exp\left(-\lambda f_1(\boldsymbol{x})\right)$$

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 $\boldsymbol{x}^*$ that maximizes the a posteriori probability $p_{\boldsymbol{X}}(\boldsymbol{x}|\boldsymbol{y})$:

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\operatorname{argmax}}\, p_{\boldsymbol{X}}(\boldsymbol{x}|\boldsymbol{y}).$$

If we assuming that $p_{\boldsymbol{X}}(\boldsymbol{x})$ is a Gibbs prior, by consider the equivalent formulation of $p_{\boldsymbol{X}}(\boldsymbol{x}|\boldsymbol{y})$ with the negative logarithm, we have:

$$\begin{aligned}\boldsymbol{x}^* &= \underset{\boldsymbol{x}\in\mathbb{R}^n}{\operatorname{argmin}} - \ln p_{\boldsymbol{X}}(\boldsymbol{x}|\boldsymbol{y}) = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\operatorname{argmin}} \left(-\ln(p_{\boldsymbol{Y}}(\boldsymbol{y}|\boldsymbol{x})) - \ln(p_{\boldsymbol{X}}(\boldsymbol{x})) + \ln(p_{\boldsymbol{Y}}(\boldsymbol{y}))\right)\\ &= \underset{\boldsymbol{x}\in\mathbb{R}^n}{\operatorname{argmin}}\, f(\boldsymbol{x};\boldsymbol{y}) \equiv f_0(\boldsymbol{x};\boldsymbol{y}) + \lambda f_1(\boldsymbol{x})\end{aligned}$$

The function f_1 is called *regularization functional*, which has the role of imposing some properties on the seeked solution; the constant λ is the *regularization parameter*, which balances the trade–off between f_0 and f_1. It is worth noting that the quality of the reconstructions obtained via a MAP approach hugely depends on the choice of the regularization parameter λ. In the classical regularization theory, a wide literature exists on the problem of the optimal choice for the regularization parameter [50].

1.3 Regularization functionals

In this section we recall some of the most classical regularizers used in image deblurring and denoising. 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:

$$\nabla_i \boldsymbol{x} = \begin{pmatrix} x_{i+1} - x_i \\ x_{i+m} - x_i \end{pmatrix}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ represents a vectorized 2D image with $n = m^2$. A similar definition is given for the Laplacian operator ∇^2.

Tikhonov regularization

Given a linear operator $A \in \mathbb{R}^{n\times n}$, the choice

$$f_1(\boldsymbol{x}) = \frac{1}{2}\|A\boldsymbol{x}\|^2$$

is known as *Tikhonov regularization* [110, 111, 112], whose starting point is the Euler equation (1.6), rising from the Least Squares formulation. The aim of this kind of regularization is to emphasize the features of smooth objects. According to the choice of A, we distinguish between different kind of Tikhonov regularizers:

- $A = I$ (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 [102] preserves discontinuities and edges in the image. The discrete version of Total Variation can be written as

$$TV(\boldsymbol{x}) = \sum_{i=1}^{n} \|\nabla_i \boldsymbol{x}\|,$$

where ∇_i is the discretization of the gradient operator at the i–th pixel. Using the notation introduced in [115], the TV functional can be expressed also as

$$TV(\boldsymbol{x}) = \sum_{i=1}^{n} \psi\Big((x_{i+1} - x_i)^2 + (x_{i+m} - x_i)^2\Big).$$

with $\psi(t) = \sqrt{t}$. The functional $TV(\boldsymbol{x})$ is convex; however, it is nondifferentiable at any point $\boldsymbol{x}$ such that $\nabla_i \boldsymbol{x} = 0$ for some $i \in \{1, \ldots, n\}$. Then, in order to remove the singularity of ψ at the origin and recover differentiability, one can introduce a generalization ψ_δ for the function ψ by means of positive threshold $\delta \in \mathbb{R}_{>0}$. Some common choices for ψ_δ can be found in [115], assuming that $\psi^{'}(t) > 0$ for $t \geq 0$:

$$\psi_\delta(t) = \sqrt{t + \delta^2} \tag{1.8}$$

$$\psi_\delta(t) = \begin{cases} \frac{t}{\delta} & \text{if } \ t \leq \delta^2 \\ 2\sqrt{t} - \delta & \text{otherwise} \end{cases} \tag{1.9}$$

The classical TV can be recovered by choosing $\delta = 0$ in (1.8); formulation (1.8) is also known as the *Hypersurface Potential* (HS) functional [115, 29]:

$$\mathrm{HS}(\boldsymbol{x}) = \sum_{i=1}^{n} \psi_\delta\Big((x_{i+1} - x_i)^2 + (x_{i+m} - x_i)^2\Big) = \sum_{i=1}^{n} \sqrt{\|\nabla_i \boldsymbol{x}\|^2 + \delta^2}.$$

where δ is considered as a thresholding parameter which tunes the value of the gradient above which a discontinuity is detected.

1.4 Constraints

In the framework of inverse problems theory it is generally accepted that, if no additional information on the object is employed, the resulting problem is ill–posed. Indeed, this is the case for the ML approach, because only information about the noise is used, with possibly the addition of nonnegativity constraints. In some applications, a priori information can be inferred from the physics underlying the acquisition process. Additional information of this kind may be employed by restricting the search of the object $\boldsymbol{x}$ onto a convex set $\Omega \subset \mathbb{R}^n$:

$$\boldsymbol{x}^* = \underset{\boldsymbol{x} \in \Omega}{\operatorname{argmin}}\, f_0(\boldsymbol{x}; \boldsymbol{y}) + f_1(\boldsymbol{x}).$$

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: $\Omega = \{\boldsymbol{x} \in \mathbb{R}^n : \sum_{i=1}^n x_i = c\}$;
- box constraint: $\Omega = \{\boldsymbol{x} \in \mathbb{R}^n : a_i \leq x_i \leq b_i, i = 1, \dots, n\}$, when some physical bounds $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$ are imposed on the object;
- box + linear inequality constraint: $\Omega = \{\boldsymbol{x} \in \mathbb{R}^n : a_i \leq x_i \leq b_i, i = 1, \dots, n, \sum_{i=1}^n x_i \leq c\}$.

Chapter 2

Forward–backward methods for constrained optimization

In the framework of this thesis we are interested in solving constrained optimization problems of the form

$$\min_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x}) \tag{2.1}$$

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 $\boldsymbol{x}$ are imposed, one speaks of *unconstrained optimization*, otherwise of *constrained optimization.*

The differentiable constrained problem (2.1) can be seen as a particular case of the more general problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \equiv f_0(\boldsymbol{x}) + f_1(\boldsymbol{x}) \tag{2.2}$$

where we keep the following hypotheses on the considered functions:

- $f_1 : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ 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 $\Omega_0 \supseteq \mathrm{dom}(f_1)$.

Formulation (2.2) corresponds to the differentiable constrained problem (2.1) when the term f_1 is chosen as the indicator function of the non empty, closed and convex set $\Omega \subset \mathbb{R}^n$, i.e.

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

The chapter starts in Section 2.1 and Section 2.2 with an overview of forward–backward methods with the aim to describe possible tools to tackle problems of type (2.1) and (2.2), followed in Section 2.3 by the illustration of possible acceleration techniques for the aforementioned methods. Finally, two particular variable metric first–order methods involved in subsequent applications are described in Section 2.4, along with their convergence results.

2.1 Mathematical tools

In this section we recall some definitions and properties of convex and variational analysis that constitute the theoretical background of successive topics. More extensive and detailed studies can be found in [98, 99, 118].

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

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

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

Definition 2.3. *[99, Definition 1.5] A function* $f : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ *is lower semicontinuous at* $\boldsymbol{x}$ *if*

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

Similarly, f *is upper semicontinuous at* $\boldsymbol{x}$ *if*

$$f(\boldsymbol{x}) = \limsup_{\boldsymbol{y} \rightarrow \boldsymbol{x}} f(\boldsymbol{x}) = \inf_{\delta > 0} \left(\sup_{\boldsymbol{y} \in B(\boldsymbol{x}, \delta)} f(\boldsymbol{y}) \right). \tag{2.4}$$

Therefore, the function f *is continuous at* $\boldsymbol{x}$ *if and only if* f *is both lower and upper semicontinuous at* $\boldsymbol{x}$*.*

Definition 2.4. *Let* D *be a symmetric positive definite* $n \times n$ *matrix and* $\Omega \subset \mathbb{R}^n$ *a nonempty, closed and convex set. The projection operator* $P_{\Omega, D} : \mathbb{R}^n \rightarrow \Omega$ *is defined as*

$$P_{\Omega, D}(\boldsymbol{x}) = \arg\min_{\boldsymbol{y} \in \Omega} \|\boldsymbol{y} - \boldsymbol{x}\|_D = \underset{\boldsymbol{y} \in \Omega}{\mathrm{argmin}} \left(\phi(\boldsymbol{y}) \equiv \frac{1}{2} \boldsymbol{y}^T D \boldsymbol{y} - \boldsymbol{y}^T D \boldsymbol{x} \right). \tag{2.5}$$

The next definition concerns the notion of proximal (or proximity) operator that was first introduced by Moreau in [83]. Here we give its most general definition with respect to a symmetric positive definite matrix.

Definition 2.5. *[55, §2.3] The* proximity *operator associated to a function* $f : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ *in the metric induced by a symmetric positive definite matrix* D *is defined as*

$$\mathrm{prox}_f^D(\boldsymbol{x}) = \arg\min_{\boldsymbol{z} \in \mathbb{R}^n} f(\boldsymbol{z}) + \frac{1}{2} \|\boldsymbol{z} - \boldsymbol{x}\|_D^2, \quad \forall \boldsymbol{x} \in \mathbb{R}^n. \tag{2.6}$$

Remark 2.1. If the matrix $D = I_n$, we denote $\mathrm{prox}_f^{I_n} = \mathrm{prox}_f$.

The operator $\mathrm{prox}_f^D : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is a multi–valued map, and for some point $\boldsymbol{x}$ one could have $\mathrm{prox}_f^D(\boldsymbol{x}) = \emptyset$. Nevertheless, existence and uniqueness of the proximal point is ensured by convexity and lower semicontinuity assumptions.

Proposition 2.1. *If $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is proper, convex and lower semicontinuous, then* $\mathrm{prox}_f^D(\boldsymbol{x})$ *exists and is unique for all $\boldsymbol{x} \in \mathbb{R}^n$.*

Proof. The function $\varphi(\boldsymbol{z}) = f(\boldsymbol{z}) + \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{x}\|_D^2$ 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. □

Example 2.1. 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 (2.5):

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

Proximity operators are therefore a generalization of projection operators.

2.2 Forward–backward methods

Forward–backward (FB) algorithms [6, 7, 32, 33] are extensively employed in order to find solutions for problems of type (2.2). The general iteration of these schemes is the following:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \lambda_k \left(\mathrm{prox}_{\alpha_k f_1}(\boldsymbol{x}^{(k)} - \alpha_k \nabla f_0(\boldsymbol{x}^{(k)})) - \boldsymbol{x}^{(k)}\right), \quad k = 0, 1, 2, \dots \tag{2.7}$$

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 linesearch) 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. Some widespread instances of (2.7) are the following ones:

- the proximal point algorithm [100] for minimizing a nondifferentiable function f_1, when $f_0 \equiv 0$ and $\lambda_k \equiv 1$:
$$\boldsymbol{x}^{(k+1)} = \mathrm{prox}_{\alpha_k f_1}(\boldsymbol{x}^{(k)});$$
- the steepest descent method, when $f_1 \equiv 0$ and $\lambda_k \equiv 1$;
- the gradient projection method, when $f_1 = \iota_\Omega$.

Each iteration (2.7) of a FB method requires the solution of the convex subproblem linked to the evaluation of the proximal operator at the gradient point. Thus, when employing forward–backward schemes to problem (2.2) we need to solve a sequence of convex subproblems whose solution needs to be expressed in closed–form or, at least, within a certain accuracy, so that the method can be effective. In the following, the proximal operator will be always assumed to be known in its exact form.

The following property is crucial for convergence analysis of forward–backward methods and it will be assumed from now on.

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

$$\|\nabla f_0(\boldsymbol{x}) - \nabla f_0(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n. \tag{2.8}$$

We now start with an overview of forward–backward methods, traditionally classified by two possible choice of linesearches, a first one in which the linesearch is performed *along the arc* and a second one which performs the linesearch *along the feasible direction.*

2.2.1 Linesearch: along the arc approach

The along the arc approach is obtained by setting $\lambda_k \equiv 1$ in (2.7):

$$\boldsymbol{x}^{(k+1)} = \mathrm{prox}_{\alpha_k f_1}\left(\boldsymbol{x}^{(k)} - \alpha_k \nabla f_0(\boldsymbol{x}^{(k)})\right). \tag{2.9}$$

The FB iteration (2.9) can be interpreted as the minimization of a suitable local approximation of the objective function. In fact, we have

$$\begin{aligned}
\boldsymbol{x}^{(k+1)} &= \mathrm{prox}_{\alpha_k f_1}\left(\boldsymbol{x}^{(k)} - \alpha_k \nabla f_0(\boldsymbol{x}^{(k)})\right) \\
&= \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} \frac{1}{2\alpha_k}\|\boldsymbol{x} - (\boldsymbol{x}^{(k)} - \alpha_k \nabla f_0(\boldsymbol{x}^{(k)}))\|^2 + f_1(\boldsymbol{x}) \\
&= \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} \underbrace{f_0(\boldsymbol{x}^{(k)}) + \nabla f_0(\boldsymbol{x}^{(k)})^T(\boldsymbol{x} - \boldsymbol{x}^{(k)}) + \frac{1}{2\alpha_k}\|\boldsymbol{x} - \boldsymbol{x}^{(k)}\|^2}_{:=q_{\alpha_k}(\boldsymbol{x})} + f_1(\boldsymbol{x}) \qquad (2.10) \\
&= \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} h_{\alpha_k}(\boldsymbol{x}).
\end{aligned}$$

Hence, at each iteration, 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 instances of the along the arc approach are illustrated by Beck and Teboulle in [6, 7]. As explained by the authors in [7, Section 1.4.2–1.4.3], when the function f_0 is convex, the convergence analysis of the along the arc scheme (2.9) is strictly related to the property stated below:

$$f_0(\boldsymbol{x}^{(k+1)}) \leq h_{\alpha_k}(\boldsymbol{x}^{(k+1)}), \quad \forall\ k \in \mathbb{N}. \tag{2.11}$$

According to (2.11), the steplength α_k need to be chosen in a way that the approximated function f_0 at the proximal point $\boldsymbol{x}^{(k+1)}$ is majorized by the local approximation h_{α_k}; the Descent Lemma stated below shows a relation between the steplength α_k to the Lipschitz constant L of ∇f_0 which will suggest possible choices for the steplength α_k.

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

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

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

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

□

If the parameter α_k is taken in the interval $(0, 1/L]$, then condition (2.11) is ensured thanks to Lemma 2.1; now, the subsequent two possibilities arise for the choice of the steplength parameter.

- If the Lipschitz constant L is known, then α_k is chosen as

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

  The method adopting this choice is known as *Iterative Soft Thresholding Algorithm (ISTA)*, where the name is borrowed from a special instance of Algorithm 1, which is recovered when $f_1 = \lambda\|\cdot\|_1$ and the proximal operator consequently reduces to the soft–thresholding operator [27, 43]; the scheme of this method is reported in Algorithm 1.

- If the Lipschitz constant L is not known or cannot be easily computed, a linesearch is performed in order to ensure condition (2.11). In particular, once the values $L_0 \in \mathbb{R}_{>0}$, $\eta > 1$ are fixed, the parameter α_k is selected as:

  $$\alpha_k = \frac{1}{L_k}, \tag{2.12}$$

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

$$f_0(\boldsymbol{x}^{(k+1)}) \leq f_0(\boldsymbol{x}^{(k)}) + (\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)})^T \nabla f_0(\boldsymbol{x}^{(k)}) + \frac{L_k}{2}\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|^2, \qquad (2.13)$$

where $\boldsymbol{x}^{(k+1)}$ is computed by means of (2.9) combined with (2.12). Thanks to Lemma 2.1, the described linesearch is well–defined, as condition (2.13) is satisfied for every $L_k \geq L$. The resulting method is known as *ISTA with backtracking* and it is resumed in Algorithm 2.

For the sake of simplicity, the following notation is introduced to indicate the proximal operator in the subsequent Algorithms:

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

Algorithm 1 ISTA with constant steplength

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

Algorithm 2 ISTA with backtracking

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

Remark 2.2. The sequence of function values $\{f(\boldsymbol{x}^{(k)})\}_{k\in\mathbb{N}}$ produced both by ISTA and ISTA with backtracking is nonincreasing. In fact, if L_k is chosen via the backtracking rule (2.13) or $L_k \equiv L$, we have:

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

where the first inequality follows from STEP 1 of Algorithm 2 and the second one is a consequence of the definition of proximal point (see the quadratic approximation interpretation 2.10).

Remark 2.3. Since (2.13) 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

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

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

Remark 2.4. The linesearch procedure (2.13) avoids the difficulty of knowing the Lipschitz constant only partially. In fact, the parameter L is linked on the value of the initial guess L_{-1} by relation $L_k = (\prod_{j=1}^{k} \eta^{i_j})L_{-1}$. As a consequence, a wrong choice of the initial guess L_{-1} might negatively affect the convergence rate of the whole algorithm. For instance, when L_{-1} is taken far from the unknown Lipschitz value, if L_{-1} is too large a very small steplength could be used or, if L_{-1} is too small, the algorithm could encounter a great number of successive linesearch reductions.
Furthermore, a critical issue of any backtracking procedure for the scheme (2.9) is that any iteration of the backtracking loop requires a new evaluation of the proximal operator. Thus, the along the arc approach becomes computationally too expensive when 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 $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ generated by the two ISTA methods to a solution of problem (2.2), along with a sublinear rate of convergence for their function values.

Theorem 2.1. *Let $f : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ be as in problem (2.2), where f_0 is convex, continuously differentiable and satisfies Assumption 2.1, and f_1 is proper, convex and lower semicontinuous. Suppose that (2.2) admits at least one solution. Let $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ be the sequence generated by Algorithm 1 or 2. Then*

(i) the sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ converges to a solution of problem (2.2).

(ii) $f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^*) = \mathcal{O}(\frac{1}{k})$ for any solution $\boldsymbol{x}^*$.

Proof. See [7, Theorem 1.1–1.2]. □

2.2.2 Linesearch: along the feasible direction approach

The second possible linesearch approach makes use of a relaxation parameter λ_k in the scheme (2.7). 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 (for example, by means of the Armijo rule [12, 89], reported in Algorithm 3). The seminal work by

Algorithm 3 Armijo linesearch algorithm

Let $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ be a sequence of points in $\mathbb{R}^n$. Choose some $\delta, \beta \in (0,1)$, $\lambda > 0$.

```
1. Set $\lambda_k = \lambda$. Let $\boldsymbol{d}^{(k)}$ be a descent direction at $\boldsymbol{x}^{(k)}$.
2. IF
       $f(\boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{d}^{(k)}) \leq f(\boldsymbol{x}^{(k)}) + \beta\lambda_k \nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{d}^{(k)}$    (2.14)
   THEN go to step 3.
   ELSE set $\lambda_k = \delta\lambda_k$ and go to step 2.
3. END
```

Combettes [33] suggested a scheme with variable steplengths variable but they strictly depend on the value of the Lipschitz constant, accordingly to the following condition:

$$0 < \inf_{k\in\mathbb{N}} \alpha_k \leq \sup_{k\in\mathbb{N}} \alpha_k < \frac{2}{L}, \tag{2.15}$$

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

$$0 < \inf_{k\in\mathbb{N}} \lambda_k \leq \sup_{k\in\mathbb{N}} \lambda_k \leq 1. \tag{2.16}$$

A special instance of this scheme has been proposed by Combettes and Pesquet in [32], here reported in Algorithm 4.

Algorithm 4 Forward–backward method with relaxation parameters and variable steplengths

Choose the starting point $\boldsymbol{x}^{(0)} \in \mathrm{dom}(f_1)$, let $L \in \mathbb{R}_{>0}$ be the Lipschitz constant of ∇f_0 and fix $\epsilon \in (0, \min\{1, 1/L\})$.

```
FOR $k = 0, 1, 2, \dots$
    STEP 1. Choose $\alpha_k \in [\epsilon, \frac{2}{L} - \epsilon]$.
    STEP 2. Compute $\boldsymbol{y}^{(k)} = \mathrm{prox}_{\alpha_k f_1}\left(\boldsymbol{x}^{(k)} - \alpha_k \nabla f_0(\boldsymbol{x}^{(k)})\right)$.
    STEP 3. Choose $\lambda_k \in [\epsilon, 1]$.
    STEP 4. Compute $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \lambda_k(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)})$.
END
```

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

Theorem 2.2. *[33, Theorem 3.4] Suppose that f_0 in problem* (2.2) *is convex, continuously differentiable and satisfies Assumption 2.1, and f_1 is proper, convex and lower semicontinuous.*

Every sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ *generated by Algorithm 4 or, more generally, by any method of type* (2.7) *satisfying conditions* (2.15)–(2.16)*, converges to a solution of problem* (2.2)*.*

Algorithm 4 features variable steplengths, but the relaxation parameters $\{\lambda_k\}_{k\in\mathbb{N}}$ cannot exceed 1. The variant proposed in [5] and resumed in Algorithm 5 allows for larger relaxation parameters, at the price of keeping fixed the steplength parameter.

Algorithm 5 Forward–backward method with relaxation parameters and constant steplength

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

Theorem 2.3. *[5] Suppose that* f_0 *in problem* (2.2) *is convex, continuously differentiable and satisfies Assumption 2.1, and* f_1 *is proper, convex and lower semicontinuous. Every sequence* $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ *generated by Algorithm 5 converges to a solution of problem* (2.2)*.*

Remark 2.5. The strategies described by Algorithm 4 and 5 can be applied only when the value of the Lipschitz constant is available. Though, for a large number of problems in signal and image processing the knowledge of the Lipschitz constant is out of reach. For instance, the Kullback–Leibler divergence with positive background (see Chapter 1, Example 1.4), 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 known [67]. In these cases, the approaches based on steplength selection by backtracking procedures may be exploited.

We now turn to the original constrained minimization problem (2.1) which is the specific goal of the analysis and the next chapters' applications and recall some notions for a simple and well studied optimization algorithm.

Classical gradient projection methods

The Gradient Projection (GP) is a method well known in literature, which exploits linesearch along the feasible direction; it is employed in the constrained optimization framework, i.e. when we consider $f_1 = \iota_\Omega$ and $\mathrm{prox}_{\iota_\Omega}(\boldsymbol{x}) = P_\Omega(\boldsymbol{x}) = \mathrm{argmin}_{\boldsymbol{z}\in\Omega} \|\boldsymbol{z} - \boldsymbol{x}\|^2$ (see Example 2.1). The

Algorithm 6 Gradient Projection (GP) method

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

general iteration of GP scheme is given by

$$\begin{aligned} \boldsymbol{x}^{(k+1)} &= \boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{d}^{(k)} = \\ &= \boldsymbol{x}^{(k)} + \lambda_k \left(P_\Omega(\boldsymbol{x}^{(k)} - \alpha_k \nabla f(\boldsymbol{x}^{(k)})) - \boldsymbol{x}^{(k)} \right), \end{aligned} \tag{2.17}$$

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. The direction $\boldsymbol{d}^{(k)}$ is a descent one at point $\boldsymbol{x}^{(k)}$ whenever $\boldsymbol{d}^{(k)} \neq 0$, otherwise $\boldsymbol{x}^{(k)}$ is a stationary point for f. The linesearch parameter λ_k is determined by means of a backtracking loop where the Armijo rule (2.14) (or its nonmonotone version (2.42)) is imposed. It is worth stressing the fact that the projection is computed only once at each iteration of the algorithm. The full GP scheme equipped with Armijo backtracking rule (2.14) is reported in Algorithm 6.

The stationarity of the limit points of the iterates of the GP method is guaranteed by the following result.

Theorem 2.4. *[12, Proposition 2.3.1] Let $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ be a sequence generated by Algorithm 6. Then every limit point of $\{\boldsymbol{x}^{(k)}\}_k$ is stationary.*

Furthermore, the convergence of the whole sequence to a solution of problem (2.1) can be proved under additional hypothesis.

Theorem 2.5. *[68, Theorem 1] Suppose that f is convex and problem* (2.1) *has at least one solution. Every sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ generated by Algorithm 6 converges to a solution of problem* (2.1).

A further hypothesis on the Lipschitz continuity of ∇f allows to evaluate the decreasing rate of the objective function, obtaining the same performance estimation proved for the ISTA methods in Theorem 2.1.

Theorem 2.6. *Assume the hypothesis of Theorem 2.5. If f has a Lipschitz continuous gradient on Ω, then*

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

Proof. See [19, Theorem 3.1][13, Proposition 6.10.2]. □

The convergence of the GP method is typically slow in practical case, so alternative forms of this scheme have been proposed in the last years [14, 15, 38, 20] aiming at accelerating its performances. In particular, the analysis of a recent variant [20] of the GP algorithm will be detailed in Section 2.4.2.

2.3 Acceleration strategies

Forward–backward methods are generally characterized by slow rate of convergence, as most of the first–order methods. Nevertheless, their practical simplicity keeps the interest on this type of schemes quite high, so that different techniques aiming to speed up their convergence can be found in literature: the addition of an extrapolation step, suitably choices of a steplength parameter and the adoption of a variable metric in the computation of the proximal operator (this latter case will be described in Section 2.4).

2.3.1 Inertial/Extrapolation techniques

Extrapolation was first introduced in gradient methods by Polyak in [90], where he studied the well–known *Heavy–Ball method* for minimizing strongly convex functions with Lipschitz continuous gradient:

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

with $\alpha \in \mathbb{R}_{>0}$, $\beta \in [0, 1)$. The term $\beta(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)})$ is usually known to as the *inertial force* or *extrapolation step*, and introduces information about the two previous iterates. The set of

Algorithm 7 FISTA with backtracking

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

the parameter $\beta = 0$ reduces to the usual gradient method. The Heavy–Ball method provides an optimal $\mathcal{O}(1/k^2)$ rate for strongly convex functions [90], with a negligible additional cost related to the extrapolation step.

The following variant of the Heavy–Ball method was initially treated by Nesterov in [87] for gradient methods and subsequently extended to forward–backward methods:

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

The two main changes with respect to the Heavy–Ball method are that the extrapolation factor β_k is variable and computed according to a prefixed formula and, furthermore, the gradient is evaluated at the extrapolated point $\boldsymbol{y}^{(k)}$ instead of $\boldsymbol{x}^{(k)}$ at each iteration. The resulting algorithm is still optimal, showing an $\mathcal{O}(1/k^2)$ complexity result for the sequence of the objective function values.

The aforementioned extrapolated scheme can be applied to forward–backward schemes in the following way:

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

The combination of Nesterov acceleration technique with the forward–backward method ISTA led to the well–known *Fast Iterative Soft Thresholding Algorithm (FISTA)* [6, 28] for solving problem (2.2) (see Algorithm 7). In the FISTA scheme, 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 selection of Beck and Teboulle in [6], namely $t_{k+1} = \left(1 + \sqrt{1 + 4t_k^2}\right)/2$, guarantees an $\mathcal{O}(1/k^2)$ convergence rate for FISTA, which improves the result contained in Theorem 2.1 for ISTA. Nevertheless, this choice for t_k does not ensure the convergence of the iterates $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$. The rule suggested by Chambolle and Dossal in [28] allows to prove the strong convergence of the sequence in $\mathbb{R}^n$, preserving the $\mathcal{O}(1/k^2)$ complexity result on the sequence of objective function values: the parameter t_k in Algorithm 7 is accordingly to the aforementioned rule. The general convergence result is here reported for the finite dimensional case.

Theorem 2.7. *[6, Theorem 4.4][28, Theorem 4.1] Let $f : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ be as in problem* (2.2)*, where f_0 is convex, continuously differentiable and satisfies Assumption 2.1, and f_1 is proper, convex and lower semicontinuous. Suppose that* (2.2) *admits at least one solution. Let $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ be the sequence generated by Algorithm 7. Then*

(i) the sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ converges to an optimal solution of problem (2.2)*.*

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

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

for any optimal solution $\boldsymbol{x}^*$.

2.3.2 Steplength selections

We will now describe some possibilities for the choice of the steplength parameter for gradients methods in the unconstrained optimization framework, which can also be extended to constrained problems, as we will detail in Section 2.4.

Barzilai–Borwein rules

Classical steplength selection rules (Minimization and Cauchy steepest descent, limited minimization [26] and Armijo (Algorithm 3)) depend on the monotonicity of the function values to guarantee global convergence of the scheme. The widely known *Barzilai–Borwein* (BB) rules [4] exploit a different procedure. In particular, the BB steplengths aim to approximate the Hessian $\nabla^2 f$ with the diagonal matrices $B(\alpha_k) = (\alpha_k I_n)^{-1}$, which are forced to satisfy one of the following quasi–Newton conditions:

$$\alpha_k^{BB1} = \underset{\alpha\in\mathbb{R}}{\operatorname{argmin}} \|B(\alpha)\boldsymbol{s}^{(k-1)} - \boldsymbol{z}^{(k-1)}\| \tag{2.18}$$

$$\alpha_k^{BB2} = \underset{\alpha\in\mathbb{R}}{\operatorname{argmin}} \|\boldsymbol{s}^{(k-1)} - B(\alpha)^{-1}\boldsymbol{z}^{(k-1)}\|, \tag{2.19}$$

where we denote $\boldsymbol{s}^{(k-1)} = \boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}$ and $\boldsymbol{z}^{(k-1)} = \nabla f(\boldsymbol{x}^{(k)}) - \nabla f(\boldsymbol{x}^{(k-1)})$. The resulting steplength values are

$$\alpha_k^{BB1} = \frac{\boldsymbol{s}^{(k-1)^T}\boldsymbol{s}^{(k-1)}}{\boldsymbol{s}^{(k-1)^T}\boldsymbol{z}^{(k-1)}} \qquad ; \qquad \alpha_k^{BB2} = \frac{\boldsymbol{s}^{(k-1)^T}\boldsymbol{z}^{(k-1)}}{\boldsymbol{z}^{(k-1)^T}\boldsymbol{z}^{(k-1)}}. \tag{2.20}$$

The choice of the approximation matrix $B(\alpha_k)$ can be motivated as follows. If we consider the Talyor expansion of $\nabla f(\boldsymbol{x}^{(k)})$

$$\nabla f(\boldsymbol{x}^{(k)}) = \nabla f(\boldsymbol{x}^{(k-1)}) + \nabla^2 f(\boldsymbol{x}^{(k-1)})\big(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\big) + o\big(\big|\big|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\big|\big|\big) \tag{2.21}$$

by discarding the terms of order higher than 1 we can write

$$\nabla^2 f(\boldsymbol{x}^{(k-1)})\underbrace{\big(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\big)}_{\boldsymbol{s}^{(k-1)}} = \underbrace{\nabla f(\boldsymbol{x}^{(k)}) - \nabla f(\boldsymbol{x}^{(k-1)})}_{\boldsymbol{z}^{(k-1)}} \tag{2.22}$$

$$\underbrace{\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}}_{\boldsymbol{s}^{(k-1)}} = \Big(\nabla^2 f(\boldsymbol{x}^{(k-1)})\Big)^{-1}\underbrace{\Big(\nabla f(\boldsymbol{x}^{(k)}) - \nabla f(\boldsymbol{x}^{(k-1)})\Big)}_{\boldsymbol{z}^{(k-1)}} \tag{2.23}$$

Thus, properties (2.18) and (2.19) force the matrix $B(\alpha_k)$ to approximate the behaviour of $\nabla^2 f(\boldsymbol{x}^{k-1})$ described in (2.22) and (2.23); this is done by exploiting available informations from the iterates and the gradient of the objective function of the current and previous step.

Numerical evidence pointed out that the BB rules and their variants are able to greatly speed up the convergence rate of several methods, for example the Cauchy steepest descent method, both in the quadratic [4, 56] and non–quadratic case [95]; Barzilai and Borwein [4] proved the R–superlinear convergence of the steepest descent method equipped with one of the steplength rule in (2.20) for two–dimensional strictly convex quadratic functions. Moreover, Raydan [95] established global convergence of the BB methods for the strictly convex quadratic case with any number of variables and, later, Dai and Liao [40] proved the R–linear convergence result. As two last results hold also for the Cauchy steepest descent, they cannot be considered a validation for BB methods efficacy.

The effectiveness of BB methods is detailed in [52, 53] for a quadratic objective function

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} - b^T\boldsymbol{x} \tag{2.24}$$

If we apply the steepest descent method for this case, we can write the gradient $\boldsymbol{g}^{(k)} = \nabla f(\boldsymbol{x}^{(k)})$ as follows

$$\begin{aligned} \boldsymbol{g}^{(k)} = A\boldsymbol{x}^{(k)} - b &= A\boldsymbol{x}^{(k-1)} - b - \alpha_{k-1}A\boldsymbol{g}^{(k-1)} \\ &= (I_n - \alpha_{k-1}A)\boldsymbol{g}^{(k-1)}. \end{aligned} \tag{2.25}$$

By applying iteratively (2.25), one obtains the following relation

$$\boldsymbol{g}^{(k)} = \left(\prod_{j=0}^{k-1}(I_n - \alpha_j A)\right) g^{(0)}. \tag{2.26}$$

A basis of eigenvectors of A will be denoted by $\{d_i\}_{i=1}^n$ and the corresponding eigenvalues by $0 < \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_{n-1} < \lambda_n$; now, the vector $g^{(0)}$ can be expressed 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 we rewrite equation (2.26) as

$$\boldsymbol{g}^{(k)} = \sum_{i=1}^n \mu_i^{(0)} \left(\prod_{j=0}^{k-1}(I_n - \alpha_j A)\right) d_i. \tag{2.27}$$

The vector $\boldsymbol{g}^{(k)}$ can also be represented with respect to the eigenvectors d_i

$$\boldsymbol{g}^{(k)} = \sum_{i=1}^n \mu_i^{(k)} d_i \tag{2.28}$$

and by comparing (2.28) with (2.27) we obtain the following relation

$$\mu_i^{(k)} = \mu_i^{(0)} \left(\prod_{j=0}^{k-1}(1 - \alpha_j \lambda_i)\right) = \mu_i^{(k-1)} \left(1 - \alpha_{k-1}\lambda_i\right). \tag{2.29}$$

The following facts can be inferred by recurrence (2.29):

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

This corresponds to the fact that, if the first n steps of the steepest descent method are defined by setting

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

then $g^{(n)} = 0$ and the method converges in at most n steps. For this reason, approximate the reciprocal of some eigenvalue of the Hessian matrix at each iteration could be a good choice for the steplength α_k. As the eigenvalues of A are usually not known exactly, the idea is to approximate them with the Rayleigh quotients of the matrix A, defined as

$$R_A(\boldsymbol{x}) = \frac{\boldsymbol{x}^T A \boldsymbol{x}}{\|\boldsymbol{x}\|^2}, \quad \forall\, \boldsymbol{x} \in \mathbb{R}^n \setminus \{0\}. \tag{2.30}$$

This approximation is justified by the fact that any eigenvalue of A is a Rayleigh quotient in which $\boldsymbol{x}$ is the corresponding eigenvector and, moreover, the minimum and maximum value of

$R_A(\boldsymbol{x})$ over $\boldsymbol{x}$ coincide with the minimum and maximum eigenvalue of A, respectively:

$$\lambda_1 = \min_{\substack{\boldsymbol{x}\in\mathbb{R}^n \\ \boldsymbol{x}\neq 0}} R_A(\boldsymbol{x}) = R_A(d_1) \tag{2.31}$$

$$\lambda_n = \max_{\substack{\boldsymbol{x}\in\mathbb{R}^n \\ \boldsymbol{x}\neq 0}} R_A(\boldsymbol{x}) = R_A(d_n) \tag{2.32}$$

Both BB steplengths can be seen as approximations of the reciprocals of the eigenvalues of A of the form (2.30), as stated by the following result.

Proposition 2.2. *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is strictly convex quadratic defined as* (2.24)*, and let $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ be generated by a gradient method of the form $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \nabla f(\boldsymbol{x}^{(k)})$. Then the BB rules can be rewritten as follows*

$$\alpha_k^{BB1} = \frac{{\boldsymbol{g}^{(k-1)}}^T \boldsymbol{g}^{(k-1)}}{{\boldsymbol{g}^{(k-1)}}^T A \boldsymbol{g}^{(k-1)}} = R_A^{-1}(\boldsymbol{g}^{(k-1)}) \tag{2.33}$$

$$\alpha_k^{BB2} = \frac{{\boldsymbol{g}^{(k-1)}}^T A \boldsymbol{g}^{(k-1)}}{{\boldsymbol{g}^{(k-1)}}^T A^2 \boldsymbol{g}^{(k-1)}} = R_A^{-1}(A^{\frac{1}{2}} \boldsymbol{g}^{(k-1)}) \tag{2.34}$$

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

$$\frac{1}{\lambda_n} \leq \alpha_k^{BB2} \leq \alpha_k^{BB1} \leq \frac{1}{\lambda_1}. \tag{2.35}$$

Proof. Since $\nabla f(\boldsymbol{x}) = A\boldsymbol{x} - b$, the relations $A\boldsymbol{s}^{(k-1)} = \boldsymbol{z}^{(k-1)}$ and $\boldsymbol{s}^{(k-1)} = -\alpha_k \boldsymbol{g}^{(k-1)}$ yield $\boldsymbol{z}^{(k-1)} = -\alpha_k A \boldsymbol{g}^{(k-1)}$. By replacing these relations in α_k^{BB1} we have

$$\alpha_k^{BB1} = \frac{\left(-\alpha_k \boldsymbol{g}^{(k-1)}\right)^T \left(-\alpha_k \boldsymbol{g}^{(k-1)}\right)}{\left(-\alpha_k \boldsymbol{g}^{(k-1)}\right)^T \left(-\alpha_k A \boldsymbol{g}^{(k-1)}\right)} = \frac{{\boldsymbol{g}^{(k-1)}}^T \boldsymbol{g}^{(k-1)}}{{\boldsymbol{g}^{(k-1)}}^T A \boldsymbol{g}^{(k-1)}}. \tag{2.36}$$

A similar process can be applied to α_k^{BB2} in order to prove (2.34). The Cauchy–Schwartz inequality implies the following relation

$${\boldsymbol{g}^{(k-1)}}^T A \boldsymbol{g}^{(k-1)} \leq \sqrt{{\boldsymbol{g}^{(k-1)}}^T \boldsymbol{g}^{(k-1)}} \sqrt{{\boldsymbol{g}^{(k-1)}}^T A^2 \boldsymbol{g}^{(k-1)}}. \tag{2.37}$$

By taking squares of both sides of (2.37) and dividing it by $({\boldsymbol{g}^{(k-1)}}^T A \boldsymbol{g}^{(k-1)}) \cdot ({\boldsymbol{g}^{(k-1)}}^T A^2 \boldsymbol{g}^{(k-1)})$ the inequality $\alpha_k^{BB2} \leq \alpha_k^{BB1}$ in (2.35) is obtained. The inequalities $\alpha_k^{BB2} \geq 1/\lambda_n$ and $\alpha_k^{BB1} \leq 1/\lambda_1$ can be inferred from the extremal properties of the Rayleigh quotient (2.31)–(2.32). □

The Cauchy steplength [26] for the Steepest Descent (SD) method can be seen as the reciprocal of a Rayleigh quotient as well. Indeed, by computing the derivative of the quadratic function with respect to α

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

and setting it to 0, we have

$$\alpha_k^{SD} = \frac{{\boldsymbol{g}^{(k)}}^T \boldsymbol{g}^{(k)}}{{\boldsymbol{g}^{(k)}}^T A \boldsymbol{g}^{(k)}} = R_A^{-1}(\boldsymbol{g}^{(k)}). \tag{2.38}$$

Nevertheless, the eigenvalues approximations provided by the sequence $\{1/\alpha_k^{BB1}\}_{k\in\mathbb{N}}$ are better than Cauchy optimal choice [53, 59] approximations: from the recurrence (2.29), we see that

$$\alpha_k \approx \frac{1}{\lambda_i} \quad \Rightarrow \quad \begin{cases} |\mu_i^{(k)}| \ll |\mu_i^{(k-1)}| \\ |\mu_j^{(k)}| < |\mu_j^{(k-1)}|, & \text{if } j < i \\ |\mu_j^{(k)}| > |\mu_j^{(k-1)}|, & \text{if } j > i, \ \lambda_j > 2\lambda_i. \end{cases} \tag{2.39}$$

For the last relation, it can be inferred that small steplengths α_k (close to $1/\lambda_n$) tend to decrease a large number of eigencomponents, while the reduction is not very effective for components corresponding to small eigenvalues. The use of large steplengths could successfully reduce those components, but this could give rise to an increment in the eigencomponents corresponding to the dominating eigenvalues, which will promote non–monotonic behaviour, both for the gradient norm and the function value. In order guarantee the monotonic behaviour, the Cauchy steplengths α_k^{SD} are expected to be small, while the reciprocals of the BB steplengths $1/\alpha_k^{BB1}$ can extend over the whole spectrum of A, with the result of forcing each component $\mu_i^{(k)}$ to go to zero.

Relation (2.39) could also suggest the need of balancing large steplengths with small ones in order to achieve effective methods. Many novel steplength selection rules, essentially relying on the alternation of Cauchy and BB steplengths, are based on this idea. For example, we can find the class of Gradient Methods with Retards (GMR) [57] which, given positive integers m and q_i, $i = 1, \dots, m$, set the steplength as follows

$$\alpha_k^{GMR} = \frac{\boldsymbol{g}_{\nu(k)}^T A^{\rho(k)-1} \boldsymbol{g}_{\nu(k)}}{\boldsymbol{g}_{\nu(k)}^T A^{\rho(k)} \boldsymbol{g}_{\nu(k)}} \tag{2.40}$$

where $\nu(k) \in \{k, k-1, \dots, \max\{0, k-m\}\}$ and $\rho(k) \in \{q_1, q_2, \dots, q_m\}$. It is worth noting that steplengths (2.33)–(2.34)–(2.38) are special instances of (2.40). In the GMR class we can find the Alternate Step (AS) gradient method [97, 37] (in which the Cauchy and BB1 steplengths are used in turns) and the Alternate Minimization (AM) method [41], where the minimization of the objective function along the line is alternated with the one–dimensional minimization of the gradient norm. These approaches are all based on a fixed alternation of the selected rules; recently, new types of rules exhibited better performances, relying on the idea of adaptively alternating the steplengths on the basis of some switching criterion, such as the Adaptive Steepest Descent (ASD) method, the Adaptive Barzilai–Borwein (ABB) method

[120] and its generalizations $\mathrm{ABB}_{\min 1}$ and $\mathrm{ABB}_{\min 2}$ [56]. In particular, the $\mathrm{ABB}_{\min 1}$ method alternates the two BB rules in the following way

$$\alpha_k^{\mathrm{ABB}_{\min 1}} = \begin{cases} \min\left\{\alpha_j^{BB2} : \ j = \max\{1, k-m\}, \dots, k\right\}, & \text{if } \frac{\alpha_k^{BB2}}{\alpha_k^{BB1}} < \tau \\ \alpha_k^{BB1}, & \text{otherwise} \end{cases} \tag{2.41}$$

where m is a nonnegative integer and $\tau \in (0,1)$. The former ABB rule is recovered when $m = 0$. The $\mathrm{ABB}_{\min 1}$ strategy generates a sequence of small steplengths with the BB2 rule, in a way that the successive value generated by the BB1 rule is a suitable approximation of the reciprocal of some small eigenvalue. The switching criterion in (2.41) is based on the relation $\alpha_k^{BB2}/\alpha_k^{BB1} = \cos^2\theta_{k-1}$, where θ_{k-1} is the angle between $A\boldsymbol{g}^{(k-1)}$ and $\boldsymbol{g}^{(k-1)}$, and allows to select the steplength α_k^{BB1}, which is the inverse of the Rayleigh quotient $R_A(\boldsymbol{g}^{(k-1)})$, only when $\boldsymbol{g}^{(k-1)}$ itself is a sufficiently good approximation of an eigenvector. The R–linear convergence can be proved for ABB, $\mathrm{ABB}_{\min 1}$ and $\mathrm{ABB}_{\min 2}$ as in [37], as they all are GMR methods; furthermore, Q–linear convergence for the error norm of the ASD method was proved in [120]. These alternating–based methods have been shown to be capable of further accelerating the convergence of the standard BB method [56].

For what concerns the non–quadratic problems, due to the non–monotonic behaviour of the sequence $\{f(\boldsymbol{x}^{(k)})\}_{k\in\mathbb{N}}$, the BB method requires a linesearch strategy that allows the objective function to increase at some iterations, and ensures global convergence of the sequence. In [96] Raydan proposed the nonmonotone linesearch technique devised by Grippo, Lampariello and Lucidi in [62], which is based on a generalization of the Armijo rule (2.14). In particular, for given scalars $\beta, \delta \in (0,1)$, $\epsilon > 1$, $\gamma > 0$, and by setting

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

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

$$f(\boldsymbol{x}^{(k)} + \delta^{m_k}\alpha_k^{(0)}\boldsymbol{d}^{(k)}) \leq f_{max} + \beta\delta^{m_k}\alpha_k^{(0)}\nabla f(\boldsymbol{x}^{(k)})^T\boldsymbol{d}^{(k)}, \tag{2.42}$$

is satisfied, where $f_{max} = \max\limits_{0\leq j\leq \min(k,M-1)} f(\boldsymbol{x}^{(k-j)})$ is the maximum value of the objective function over the last M iterations, being M a prefixed positive integer. The standard Armijo rule (2.14) is recovered for M equal to 1. This results in a globally convergent scheme (denominated Global Barzilai and Borwein (GBB) algorithm), as each limit point of its sequence is stationary for the objective function [96, Theorem 2.1].

Ritz values based rule

We will now describe a different approach for steplength selection rules, proposed by Fletcher [54] in the context of steepest descent methods for the quadratic objective function (2.24). This limited–memory scheme employs the most recent m back gradients

$$G = \left[\boldsymbol{g}^{(k-m)} \; \dots \; \boldsymbol{g}^{(k-2)} \quad \boldsymbol{g}^{(k-1)}\right] \tag{2.43}$$

to define the next m steplengths α_{k+i-1}, $i = 1, \dots, m$. If we apply iteratively (2.25) to the vector $\boldsymbol{g}^{(k-i)}$ for $m - i$ times, we have

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

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

$$\left\{\boldsymbol{g}^{(k-m)},\ A\boldsymbol{g}^{(k-m)},\ A^2\boldsymbol{g}^{(k-m)}, \dots,\ A^{(m-1)}\boldsymbol{g}^{(k-m)}\right\}. \tag{2.44}$$

This property allows to write the matrix G by exploiting a special basis of the Krylov sequence provided by the Lanczos iterative process [60] applied to the matrix A with starting vector $\boldsymbol{q}_1 = \boldsymbol{g}^{(k-m)}/\|\boldsymbol{g}^{(k-m)}\|$. Indeed, the Lanczos process generates an orthonormal basis $\{\boldsymbol{q}_1, \boldsymbol{q}_2, \dots, \boldsymbol{q}_m\}$ for the Krylov sequence (2.44) and the matrix G can be written as $G = QR$, where Q is the $n \times m$ orthogonal matrix with columns $\boldsymbol{q}_1, \boldsymbol{q}_2, \dots, \boldsymbol{q}_m$ and R is an $m \times m$ upper triangular matrix which is non singular, provided that the columns of G are linearly independent. Furthermore, the columns of the matrix Q are generated by the Lanczos process in such a way that

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

is a tridiagonal matrix; the eigenvalues of the matrix Φ are called *Ritz values* and provide special approximations of the eigenvalues of the matrix A. If $m = n$, the Ritz values θ_i, $i = 1, \dots, m$, coincide with the eigenvalues of A while, if $m = 1$, then $Q = \boldsymbol{q}_1 = \boldsymbol{g}^{(k-m)}/\|\boldsymbol{g}^{(k-m)}\|$ and there is a unique Ritz value, i.e. the Rayleigh quotient $R_A(\boldsymbol{g}^{(k-1)})$ on which the BB method is based. For a general m, the Ritz values lie in the spectrum of A, as each one of them can be seen as the Rayleigh quotient $\theta_i = R_A(Q\boldsymbol{y}_i)$ in which $\boldsymbol{y}_i$ 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 $m \to n$ [65].

The Limited Memory Steepest Descent (LMSD) proposed by Fletcher divides the sequence of the steepest descent method into groups of m iterations denominated *sweeps*, and selects the next m steplengths for the current sweep as the reciprocals of the m Ritz values available from the previous sweep, namely

$$\boldsymbol{x}^{(k+i)} = \boldsymbol{x}^{(k+i-1)} - \alpha_{k+i-1}\boldsymbol{g}^{(k+i-1)}, \quad i = 1, \dots, m \tag{2.45}$$

where $\alpha_{k+i-1} = (\theta_{k+i-1})^{-1}$. The convergence of the LMSD algorithm is proved in the quadratic case [54], following the arguments in [95] for the BB method; recently, R–linear convergence for this scheme has been proved in [34]. It is worth noticing that the Ritz values can be computed without explicitly use of the matrices A and Q. This allows to reduce the computational time of the LMSD method and to extend the rule to the non–quadratic case, where the matrix A is not available. Indeed, by rewriting equation (2.25) as follows

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

then it can be rearranged in the matrix form

$$AG = [G \quad \boldsymbol{g}^{(k)}]\Gamma \tag{2.46}$$

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

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

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

$$\Phi = [R \quad r]\Gamma R^{-1}. \tag{2.47}$$

Now one can compute the Cholesky factorization $G^T G = R^T R$ and solve the upper triangular linear system $R^T r = G^T\boldsymbol{g}^{(k)}$ before computing the tridiagonal matrix Φ by means of equation (2.47), in which the matrices A and Q do not appear.

In the case of a general objective function, the matrix 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

$$\widetilde{\Phi} = \mathrm{diag}(\Phi) + \mathrm{tril}(\Phi, -1) + \mathrm{tril}(\Phi, -1)^T, \tag{2.48}$$

where diag$(\cdot)$ and tril$(\cdot, -1)$ denote the diagonal and the strictly lower triangular parts of a matrix, respectively. Even though negative eigenvalues of the resulting matrix could arise, they are discarded from the choice steplengths for the next iterations. Numerical evidence [54] shows that the LMSD method outperforms the standard Barzilai–Borwein scheme for both quadratic and non–quadratic test problems, and the algorithm is competitive with other state–of–the–art methods, such as the BFGS method or the nonlinear Conjugate Gradient (CG) methods. Furthermore, in [48] the LMSD approach well compared with efficient alternating BB rules like $\mathrm{ABB}_{\min 1}$.

Other state–of–the–art steplength selection rules

An important feature of the previous steplength selections is that they can be applied not only to quadratic problems but also in the case of more general nonlinear optimization problems. However, the recent literature shows that there are other promising steplength rules designed for quadratic problems that behave similarly to the BB or Ritz based rules and deserve to be investigated for possible extension to the general non–quadratic case. Examples of such steplength rules are the selections SDA and SDC proposed in [44, 45]. The strategy devised by these methods aims to exploit SD steplengths without asymptotically limiting the search to two–dimensional space, as in the SD method; thus, a certain number of previously computed SD steplengths is followed by some constant steplengths, as follows. Given two integers $h \geq 2$ and $m_c \geq 1$, the SDA and SDC steplength are computed as

$$\alpha_k = \begin{cases} \alpha_k^{\mathrm{SD}} & \text{if } \bmod(k, h+m_c) < h, \\ \hat{\alpha}_s & \text{otherwise, with } s = \max\{i \leq k \,:\, \bmod(i, h+m_c) = h\}, \end{cases} \tag{2.49}$$

where $\hat{\alpha}_s$ is a particular steplength built at iteration s by means of $\alpha_{s-1}^{\mathrm{SD}}$ and α_s^{SD}. Strictly speaking, the methods make h consecutive exact linesearches and then compute a different steplength, which is kept constant and applied in m_c consecutive iterations. In the SDA method $\hat{\alpha}_s = \alpha_s^{\mathrm{A}}$, where

$$\alpha_s^{\mathrm{A}} = \left(\frac{1}{\alpha_{s-1}^{\mathrm{SD}}} + \frac{1}{\alpha_s^{\mathrm{SD}}} \right)^{-1},$$

while in the SDC method $\hat{\alpha}_s = \alpha_s^{\mathrm{Y}}$, where

$$\alpha_s^{\mathrm{Y}} = 2 \left(\sqrt{ \left(\frac{1}{\alpha_{s-1}^{\mathrm{SD}}} - \frac{1}{\alpha_s^{\mathrm{SD}}} \right)^2 + 4 \frac{\|\boldsymbol{g}_s\|^2}{\left(\alpha_{s-1}^{\mathrm{SD}} \|\boldsymbol{g}_{s-1}\|\right)^2} } + \frac{1}{\alpha_{s-1}^{\mathrm{SD}}} + \frac{1}{\alpha_s^{\mathrm{SD}}} \right)^{-1}. \tag{2.50}$$

It is worth noticing that α_s^{Y} is the so–called Yuan steplength [117], used in the Dai–Yuan method. This method alternates some Cauchy steplengths with some Yuan steplengths similarly to (2.49), with the difference that α_s^{Y} is computed at each iteration and it is not constant.

The numerical study reported in [48] shows that the SDC rule shares with the $\mathrm{ABB}_{\min 1}$ and the Ritz rules the ability to efficiently approximate the eigenvalues of the Hessian of the quadratic objective function, achieving a better performance with respect to the standard BB1 approach.

2.4 Variable metric techniques

In order to enhance the convergence rate, the general iteration of of forward–backward methods can be modified by the introduction of a variable metric in the computation of the proximity operator, as follows: if D_k is a symmetric positive definite matrix, we have

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \lambda_k \left(\operatorname{prox}^{D_k}_{\alpha_k f_1}(\boldsymbol{x}^{(k)} - \alpha_k D_k^{-1}\nabla f_0(\boldsymbol{x}^{(k)})) - \boldsymbol{x}^{(k)}\right), \quad k = 0, 1, 2, \ldots \tag{2.51}$$

We will refer to this scheme as the *Variable Metric Forward Backward* (VMFB) algorithm. As already pointed out in Section 2.2.1 for the case $D_k = I_n$, the variable metric forward–backward step can be seen as the minimization of a local approximation of f at the iterate $\boldsymbol{x}^{(k)}$:

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

As a consequence, the scaling matrix D_k needs to be chosen in a way that the quadratic model $q(\boldsymbol{y}, \boldsymbol{x}^{(k)})$ represents a better approximation than the one defined by the FB method in (2.10), in order to effectively improve the performances of the FB scheme by the variable metric. For example, under the hypothesis that f_0 is twice continuously differentiable and D_k approximates the Hessian matrix $\nabla^2 f_0(\boldsymbol{x}^{(k)})$, the quadratic term $q(\boldsymbol{y}, \boldsymbol{x}^{(k)})$ can be close to the second order Taylor expansion of the function f_0 at point $\boldsymbol{x}^{(k)}$.

The problem of devising effective and practical techniques to compute the scaling matrix D_k will be discussed in the following sections.

2.4.1 Gradient Projection methods with extrapolation and scaling matrix

In this section we describe a variable metric forward–backward method with extrapolation presented in [17] for the solution of problem (2.2) under the following hypotheses, which will be assumed in this section.

(H1) $f_0, f_1 : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ are proper, convex and lower semicontinuous;

(H2) f_0 is differentiable with L–Lipschitz continuous gradient on the closed and convex set $Y \subseteq \operatorname{dom}(f_0)$ and $\operatorname{dom}(f_0) \supseteq Y \supseteq \operatorname{dom}(f_1)$;

Algorithm 8 Scaled inertial forward–backward method with backtracking

```
Choose $\alpha_0 > 0$, $\delta < 1$, $\boldsymbol{x}^{(0)} \in Y$. Set $\boldsymbol{x}^{(-1)} = \boldsymbol{x}^{(0)}$ and define a sequence of nonnegative numbers
$\{\beta_k\}$ and a sequence of operators $\{D_k\}$, with $D_k \in \mathcal{M}_\eta$, $\eta \geq 1$, such that $\gamma = \sup_{k\in\mathbb{N}}\|D_k\| < \infty$.
FOR $k = 0, 1, 2, \ldots$

    STEP 1. Extrapolation: $\boldsymbol{y}^{(k)} = P_{Y,D_k}(\boldsymbol{x}^{(k)} + \beta_k(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}))$.

    STEP 2. Set $\alpha_k = \alpha_{k-1}$, $i_k = 0$.

    STEP 3. Set $\boldsymbol{x}_+^{(k)} = \mathrm{prox}_{\alpha_k g}^{D_k}(\boldsymbol{y}^{(k)} - \alpha_k {D_k}^{-1}\nabla f_0(\boldsymbol{y}^{(k)}))$.

    STEP 4. Backtracking loop:
            IF $f(\boldsymbol{x}_+^{(k)}) \leq f_0(\boldsymbol{y}^{(k)}) + \langle \nabla f_0(\boldsymbol{y}^{(k)}), \boldsymbol{x}_+^{(k)} - \boldsymbol{y}^{(k)}\rangle + \frac{1}{2\alpha_k}\|\boldsymbol{y}^{(k)} - \boldsymbol{x}_+^{(k)}\|_{D_k}^2$ THEN
                go to STEP 5
            ELSE
                $i_k \leftarrow i_k + 1 \quad \alpha_k = \delta^{i_k}\alpha_{k-1}$ and go to STEP 3.
            ENDIF

    STEP 5. Set $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}_+^{(k)}$.

END
```

(H3) problem (2.2) admits at least a solution.

The generic scheme of the method is detailed in Algorithm 8, where $\mathcal{M}_\eta$ denotes the set of all symmetric positive definite matrices with eigenvalues contained in the interval $[\frac{1}{\eta}, \eta]$ for a $\eta \geq 1$.

This scheme combines a variable metric forward–backward iteration with an extrapolation–projection step; the steplength parameter α_k is adaptively computed by means of a backtracking procedure, while the extrapolation parameter β_k has the form

$$\beta_k = \frac{\theta_k(1-\theta_{k-1})}{\theta_{k-1}}, \quad \beta_0 = 0 \ , \tag{2.52}$$

where $\{\theta_k\} \subset (0, 1]$ is a given sequence of parameters. Suitable choices for the scaling operator D_k will be described in the following.

Algorithm 8 can be considered a generalization of FISTA method described in Algorithm 7: it enhances the classical FISTA scheme by the introduction of the variable metric induced by the scaling matrices D_k at each iteration and the projection of the extrapolated point $\boldsymbol{x}^{(k)} + \beta_k(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)})$, which allows to handle problems where $\mathrm{dom}(f_0) \supseteq Y$ does not coincide with the entire space $\mathbb{R}^n$. When $Y = \mathbb{R}^n$, FISTA is recovered by setting $D_k = I_n$ for all $k \geq 0$.

Algorithm 8 is well defined thanks to assumption (H2), i.e., the backtracking loop terminates in a finite number of steps. In fact, as the sequence $\{\alpha_k\}$ is non–increasing and the reducing factor is $\delta < 1$, the following inequalities can be deduced from Lemma 2.1

$$0 < \frac{\delta\eta}{L} \leq \alpha_k \leq \alpha_{k-1} \leq \alpha_0 \,. \tag{2.53}$$

The backtracking condition implies that the new iterate $\boldsymbol{x}^{(k+1)}$ satisfies

$$f(\boldsymbol{x}^{(k+1)}) \leq f_0(\boldsymbol{y}^{(k)}) + \langle \nabla f_0(\boldsymbol{y}^{(k)}), \boldsymbol{x}_+^{(k)} - \boldsymbol{y}^{(k)} \rangle + \frac{1}{2\alpha_k} \|\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k+1)}\|_{D_k}^2 \,. \tag{2.54}$$

The convergence rate of Algorithm 8 with respect to the objective function values remains the same of the original FISTA scheme, i.e.

$$f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^*) = \mathcal{O}\left(\frac{1}{k^2}\right),$$

as proved in the results stated in the next paragraph; moreover, the sequence of the iterates $\{\boldsymbol{x}^{(k)}\}$ generated by Algorithm 8 weakly converges to a minimizer of problem (2.2).

Convergence analysis

In this paragraph we will denote by $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ the sequence generated by Algorithm 8 and $\boldsymbol{x}^*$ any of the solutions of (2.2).

We report here the result that establishes the convergence rate of the method of interest.

Theorem 2.8. *[17, Theorem 12] Let $\{D_k\} \subset \mathcal{M}_\eta$ be a sequence of operators satisfying*

$$D_{k+1} \preceq (1+\eta_k) D_k \quad \forall k \geq 0 \quad \text{with } \eta_k \in \mathbb{R}, \eta_k \geq 0 \text{ such that } \sum_{k=0}^{\infty} \eta_k < \infty \,. \tag{2.55}$$

and assume that $\{\theta_k\}$, $\{\beta_k\}$ are chosen as

$$\theta_k = \begin{cases} 1 & k = -1, 0 \\ \frac{a}{k+a} & k \geq 1 \end{cases} \qquad \beta_k = \begin{cases} 0 & k = 0 \\ \frac{k-1}{k+a} & k \geq 1 \end{cases} \tag{2.56}$$

with $a \geq 2$. Then, there exists a constant C such that

$$f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^*) \leq \frac{C}{(k-1+a)^2} \,, \tag{2.57}$$

for all $k \geq 1$. In particular $C = \frac{a^2 L \prod_{i=1}^{k-1}(1+\eta_i)\|\boldsymbol{x}^* - \boldsymbol{x}^{(0)}\|_{D_1}^2}{2\eta\delta}$.

The following result shows the convergence of the iterates to a minimizer of the function f.

Theorem 2.9. *[17, Theorem 17] Assume that $\{\theta_k\}$ and $\{\beta_k\}$ are chosen as in* (2.56) *with $a > 2$ and let $\{D_k\} \subset \mathcal{M}_\eta$ be a sequence of operators satisfying* (2.55) *and*

$$D_k \preceq (1+\nu_k)D_{k+1} \quad \forall k \geq 0 \quad \textit{with } \nu_k \in \mathbb{R},\ \nu_k \geq 0 \textit{ such that } \sum_{k=0}^{\infty} \nu_k < \infty\,. \tag{2.58}$$

with $\sup_{k\in\mathbb{N}} \|D_k\| = \gamma < \infty$, $\{\eta_k\} = \mathcal{O}(\frac{1}{k^p})$ and $\{\nu_k\} = \mathcal{O}(\frac{1}{k^p})$ with $p > 2$. Then, the sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ weakly converges to a minimizer of f.

Scaling matrix

We report here some hints for the practical choices of proper sequences $\{D_k\}$ of scaling matrices, presenting ideas which could guide the adoption of a specific metric.

- The choice of the scaling matrix should be oriented to the improvement of convergence speed without introducing significant computational costs: for this reason, diagonal scaling matrices usually represent a good compromise. Furthermore, the variable metric should record information of some kind about the problem, thus it strongly depends on the structure of the objective function and on the solution constraints, if any.
- Given a sequence $\{D_k\}$ of symmetric and positive definite matrices of order n, the following sufficient conditions guarantee both (2.55) and (2.58)

  $$\begin{cases} \|D_k^{-1}\| \leq \gamma_k \\ \|D_k\| \leq \gamma_k \end{cases}, \quad \gamma_k^2 = 1+\zeta_k \quad \text{where} \quad \zeta_k \geq 0 \quad \text{and} \quad \sum_{k=0}^{\infty} \zeta_k < \infty \tag{2.59}$$

  with $\gamma_k < \gamma$, $\gamma > 1$.

  Thus, for the practical case of diagonal scaling matrices, conditions (2.59) allow to bound the matrices' elements by means of suitable sequences of parameters $\{\zeta_k\}_{k\in\mathbb{N}}$.

2.4.2 Scaled Gradient Projection methods

The Scaled Gradient Projection (SGP) method was first presented in [20] as an extension of the GP method (2.17), which is based on the subsequent iteration

$$\begin{aligned} \boldsymbol{x}^{(k+1)} &= \boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{d}^{(k)} = \\ &= \boldsymbol{x}^{(k)} + \lambda_k \left(P_{\Omega,D_k}(\boldsymbol{x}^{(k)} - \alpha_k D_k^{-1} \nabla f(\boldsymbol{x}^{(k)})) - \boldsymbol{x}^{(k)} \right), \end{aligned} \tag{2.60}$$

with the following setting:

- α_k is a positive steplength chosen in the bounded interval $[\alpha_{min}, \alpha_{max}]$;

Algorithm 9 Scaled Gradient Projection (SGP) method

```
Choose the starting point $\boldsymbol{x}^{(0)} \in \Omega$, set the parameters $\beta, \delta \in (0,1)$, $0 < \alpha_{min} < \alpha_{max}$ and $\mu \geq 1$.
FOR $k = 0, 1, 2, \dots$
    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 $\boldsymbol{y}^{(k)} = P_{\Omega, D_k}(\boldsymbol{x}^{(k)} - \alpha_k D_k^{-1} \nabla f(\boldsymbol{x}^{(k)}))$;
            if $\boldsymbol{y}^{(k)} = \boldsymbol{x}^{(k)}$, then $\boldsymbol{x}^{(k)}$ is a stationary point and SGP stops.
    STEP 3. Define the descent direction $\boldsymbol{d}^{(k)} = \boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}$.
    STEP 4. Set $\lambda_k = 1$.
    STEP 5. Backtracking loop:
            IF $f(\boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{d}^{(k)}) \leq f(\boldsymbol{x}^{(k)}) + \beta \lambda_k \nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{d}^{(k)}$ THEN
                go to STEP 6
            ELSE
                set $\lambda_k = \delta \lambda_k$ and go to STEP 5.
            ENDIF
    STEP 6. Set $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{d}^{(k)}$.
END
```

- D_k is a symmetric positive definite matrix whose eigenvalues lie in the bounded interval $[\frac{1}{\mu_k}, \mu_k]$ with $\mu \leq \mu_k \geq 1$;
- the linesearch parameter $\lambda_k \in (0, 1]$ is determined along the feasible direction by imposing the Armijo rule (2.14).

The linesearch parameter λ_k can be determined also by imposing nonmonotone rules [20], but these cases won't be discussed in this thesis. The general SGP scheme is reported in Algorithm 9 and it differs from the original GP algorithm by the possibility to employ adaptive strategies related to the choice of the steplength parameter α_k and of the scaling matrix D_k. In the last few years, the effectiveness of the SGP method has been extensively tested in a variety of image reconstruction problems arising in microscopy and astronomy frameworks [9, 16, 18, 79, 91, 92].

Convergence analysis

We report here some useful propositions and lemmas that allow to prove the most recent results on SGP convergence analysis and recall a property which will be used in the following. In this paragraph we will denote by $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ the sequence generated by Algorithm 9 and X^* the set of the solutions of (2.2).

Remark 2.6. For any symmetric positive definite $D \in \mathcal{M}_\mu$ we have that D^{-1} also belongs to $\mathcal{M}_\mu$ and

$$\frac{1}{\mu}\|\boldsymbol{x}\|^2 \leq \|\boldsymbol{x}\|_D^2 \leq \mu\|\boldsymbol{x}\|^2 \qquad \forall \boldsymbol{x} \in \mathbb{R}^n. \tag{2.61}$$

Lemma 2.2. *[19, Lemma 2.1] Let $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ be a sequence of points in Ω and $\{\boldsymbol{d}^{(k)}\}_{k\in\mathbb{N}}$ be a sequence of descent directions such that $\nabla f(\boldsymbol{x}^{(k)})^T\boldsymbol{d}^{(k)} < 0$ $\forall k \in \mathbb{N}$. Suppose that there exists $l \in \mathbb{R}$ such that $f(\boldsymbol{x}) \geq l$ for all $\boldsymbol{x} \in \Omega$ and that*

$$f(\boldsymbol{x}^{(k+1)}) \leq f(\boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{d}^{(k)}) \quad \forall k \in \mathbb{N}. \tag{2.62}$$

Then we have

$$0 \leq -\sum_{k=0}^{\infty} \lambda_k \nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{d}^{(k)} < \infty. \tag{2.63}$$

Theorem 2.10. *[19, Theorem 2.1] Let $\alpha_{min}, \alpha_{max}, \mu$ be three positive constants such that $0 < \alpha_{min} \leq \alpha_{max}$ and $\mu \geq 1$. Let $\{\alpha_k\}_{k\in\mathbb{N}} \subset [\alpha_{min}, \alpha_{max}]$ be a sequence of parameters and $\{D_k\}_{k\in\mathbb{N}} \subset \mathcal{M}_\mu$. Let $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}} \subset \Omega$ be any sequence satisfying property (2.62), where $\boldsymbol{d}^{(k)}$ is defined in (2.60) and λ_k is computed with the Armijo linesearch procedure (2.14). If $\bar{\boldsymbol{x}}$ is a limit point of $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$, then $\bar{\boldsymbol{x}}$ is a stationary point for problem (2.1).*

Proposition 2.3. *[19, Proposition 2.2] Assume that ∇f satisfies one of the following conditions:*

a) ∇f is globally Lipschitz on Ω;

b) ∇f is locally Lipschitz and the set $\{\boldsymbol{x} \in \Omega : f(\boldsymbol{x}) \leq \zeta\}$ is bounded for every $\zeta \in \mathbb{R}$.

Let $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ be any sequence satisfying the assumptions of Theorem 2.10 and $\{\lambda_k\}_{k\in\mathbb{N}}$ the related steplengths computed by Algorithm 3. Then, there exists a positive constant $0 < \lambda_{\min} \leq 1$ such that

$$\lambda_k \geq \lambda_{\min}. \tag{2.64}$$

Lemma 2.3. *[19, Lemma 3.1] Let $\{\mu_k\}_{k\in\mathbb{N}}$, $\{\zeta_k\}_{k\in\mathbb{N}}$ be two sequences of numbers such that*

$$\mu_k^2 = 1 + \zeta_k, \quad \zeta_k \geq 0, \quad \sum_{k=0}^{\infty} \zeta_k < \infty. \tag{2.65}$$

Then the sequence $\{\theta_k\}_{k\in\mathbb{N}}$, with $\theta_k = \prod_{j=0}^{k} \mu_j^2$, is bounded.

Convergence results of the SGP sequence to a solution of (2.1) were recently proved in the convex case [19] by extending the result in [68], with the assumption that the scaling matrices D_k asymptotically reduce to the identity matrix. Such a requirement can be expressed in terms of the bounds of the eigenvalues $\{\mu_k\}_{k\in\mathbb{N}}$, as reported in the following.

Theorem 2.11. *[19, Theorem 3.1] Assume that the objective function of* (2.1) *is convex and the solution set X^* is not empty. Let $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ be the sequence generated by SGP where $D_k \in \mathcal{M}_{\mu_k}$ and $\{\mu_k\}_{k\in\mathbb{N}}$ satisfies* (2.65)*. Then the sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ converges to a solution of* (2.1)*.*

Proof. We recall the following basic norm equality, which holds true for any positive definite matrix E:

$$\|\boldsymbol{x}-\boldsymbol{y}\|_E^2 + \|\boldsymbol{y}-\boldsymbol{z}\|_E^2 - \|\boldsymbol{x}-\boldsymbol{z}\|_E^2 = 2(\boldsymbol{y}-\boldsymbol{x})^T E(\boldsymbol{y}-\boldsymbol{z}) \ . \tag{2.66}$$

Let $\hat{\boldsymbol{x}} \in X^*$. Recalling the definition of $\boldsymbol{y}^{(k)}$ in STEP 2 as $\boldsymbol{y}^{(k)} = P_{\Omega,D_k}(\boldsymbol{x}^{(k)} - \alpha_k D_k^{-1}\nabla f(\boldsymbol{x}^{(k)}))$, from the first–order necessary condition we have that

$$(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)} + \alpha_k D_k^{-1}\nabla f(\boldsymbol{x}^{(k)}))^T D_k(\boldsymbol{x}-\boldsymbol{y}^{(k)}) \geq 0 \quad \forall \boldsymbol{x}\in\Omega \ .$$

If we take $\boldsymbol{x} = \hat{\boldsymbol{x}}$, by adding and substracting the quantities $(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)})^T D_k \boldsymbol{x}^{(k)}$ and $\alpha_k \nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{x}^{(k)}$ to the previous relation we have

$$(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)})^T D_k(\hat{\boldsymbol{x}} - \boldsymbol{x}^{(k)}) \geq \alpha_k \nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{x}^{(k)} - \hat{\boldsymbol{x}}) + (\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)} + \alpha_k D_k^{-1}\nabla f(\boldsymbol{x}^{(k)}))^T D_k(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}).$$

The convexity of the function f implies $f(\hat{\boldsymbol{x}}) \geq f(\boldsymbol{x}^{(k)}) + \nabla f(\boldsymbol{x}^{(k)})^T D_k(\hat{\boldsymbol{x}} - \boldsymbol{x}^{(k)})$, so that $\nabla f(\boldsymbol{x}^{(k)})^T D_k(\boldsymbol{x}^{(k)} - \hat{\boldsymbol{x}}) \geq f(\boldsymbol{x}^{(k)}) - f(\hat{\boldsymbol{x}})$ and

$$\begin{aligned} &\alpha_k \nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{x}^{(k)} - \hat{\boldsymbol{x}}) + (\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)} + \alpha_k D_k^{-1}\nabla f(\boldsymbol{x}^{(k)}))^T D_k(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}) \\ \geq\ &\alpha_k(f(\boldsymbol{x}^{(k)}) - f(\hat{\boldsymbol{x}})) + \|\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}\|_{D_k}^2 + \alpha_k \nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}); \end{aligned}$$

the definition of $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \lambda_k(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)})$ yields

$$\begin{aligned} &\alpha_k(f(\boldsymbol{x}^{(k)}) - f(\hat{\boldsymbol{x}})) + \|\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}\|_{D_k}^2 + \alpha_k \nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}) \\ =\ &\alpha_k(f(\boldsymbol{x}^{(k)}) - f(\hat{\boldsymbol{x}})) + \frac{1}{{\lambda_k}^2}\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|_{D_k}^2 + \alpha_k \nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}). \end{aligned}$$

Thus, the following relation is now proved:

$$(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)})^T D_k(\hat{\boldsymbol{x}} - \boldsymbol{x}^{(k)}) \geq\ \alpha_k(f(\boldsymbol{x}^{(k)}) - f(\hat{\boldsymbol{x}})) + \frac{1}{{\lambda_k}^2}\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|_{D_k}^2 + \alpha_k \nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}).$$

Now, from the definition of $\boldsymbol{x}^{(k+1)}$ and by applying equality (2.66) with $\boldsymbol{x} = \boldsymbol{x}^{(k+1)}$, $\boldsymbol{y} = \boldsymbol{x}^{(k)}$, $\boldsymbol{z} = \hat{\boldsymbol{x}}$ and $E = D_k$, we can obtain

$$\begin{aligned} \|\boldsymbol{x}^{(k+1)} - \hat{\boldsymbol{x}}\|_{D_k}^2 &= \|\boldsymbol{x}^{(k)} - \hat{\boldsymbol{x}}\|_{D_k}^2 + \|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|_{D_k}^2 - 2(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k+1)})^T D_k(\boldsymbol{x}^{(k)} - \hat{\boldsymbol{x}}) \\ &= \|\boldsymbol{x}^{(k)} - \hat{\boldsymbol{x}}\|_{D_k}^2 + \|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|_{D_k}^2 - 2\lambda_k(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)})^T D_k(\hat{\boldsymbol{x}} - \boldsymbol{x}^{(k)}). \end{aligned}$$

From the convexity of the function f and from the fact that the point $\hat{\boldsymbol{x}} \in X^*$ is a solution, we have that

$$\begin{aligned}
&\|\boldsymbol{x}^{(k)} - \hat{\boldsymbol{x}}\|_{D_k}^2 + \|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|_{D_k}^2 - 2\lambda_k(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)})^T D_k(\hat{\boldsymbol{x}} - \boldsymbol{x}^{(k)}) \\
\leq\ & \|\boldsymbol{x}^{(k)} - \hat{\boldsymbol{x}}\|_{D_k}^2 + \left(1 - \frac{2}{\lambda_k}\right)\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|_{D_k}^2 - 2\alpha_k\lambda_k\nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}) + \\
& -2\lambda_k\alpha_k(f(\boldsymbol{x}^{(k)}) - f(\hat{\boldsymbol{x}}))
\end{aligned}$$

As $\lambda_k \leq 1$ and $\hat{\boldsymbol{x}} \in X^*$, the previous relations result in

$$\begin{aligned}
\|\boldsymbol{x}^{(k+1)} - \hat{\boldsymbol{x}}\|_{D_k}^2 &\leq \|\boldsymbol{x}^{(k)} - \hat{\boldsymbol{x}}\|_{D_k}^2 - 2\alpha_k\lambda_k\nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}) + \\
&\quad -2\lambda_k\alpha_k(f(\boldsymbol{x}^{(k)}) - f(\hat{\boldsymbol{x}})) \qquad (2.67)\\
&\leq \|\boldsymbol{x}^{(k)} - \hat{\boldsymbol{x}}\|_{D_k}^2 - 2\alpha_k\lambda_k\nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}).
\end{aligned}$$

From the last inequality and from relation (2.61), it follows that

$$\begin{aligned}
\frac{1}{\mu_k}\|\boldsymbol{x}^{(k+1)} - \hat{\boldsymbol{x}}\|^2 &\leq \|\boldsymbol{x}^{(k+1)} - \hat{\boldsymbol{x}}\|_{D_k}^2 \\
&\leq \|\boldsymbol{x}^{(k)} - \hat{\boldsymbol{x}}\|_{D_k}^2 - 2\alpha_k\lambda_k\nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}) \\
&\leq \mu_k\|\boldsymbol{x}^{(k)} - \hat{\boldsymbol{x}}\|^2 - 2\alpha_k\lambda_k\nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}),
\end{aligned}$$

so that

$$\|\boldsymbol{x}^{(k+1)} - \hat{\boldsymbol{x}}\|^2 \leq \mu_k^2\|\boldsymbol{x}^{(k)} - \hat{\boldsymbol{x}}\|^2 - 2\mu_k\alpha_k\lambda_k\nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}).$$

Since $\mu_k \geq 1$ and $\alpha_k \leq \alpha_{max}$, the following relation holds true:

$$\|\boldsymbol{x}^{(k+1)} - \hat{\boldsymbol{x}}\|^2 \leq \mu_k^2\|\boldsymbol{x}^{(k)} - \hat{\boldsymbol{x}}\|^2 - 2\alpha_{max}\mu_k^2\lambda_k\nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}).$$

By repeatedly applying the previous inequality we obtain

$$\|\boldsymbol{x}^{(k+1)} - \hat{\boldsymbol{x}}\|^2 \leq \theta_0^k\|\boldsymbol{x}^{(0)} - \hat{\boldsymbol{x}}\|^2 - 2\alpha_{max}\sum_{j=0}^{k}\theta_j^k\lambda_j\nabla f(\boldsymbol{x}^{(j)})^T(\boldsymbol{y}^{(j)} - \boldsymbol{x}^{(j)}),$$

where $\theta_j^k = \prod_{i=j}^k \mu_j^2$. As $\mu_j^2 \geq 1$, it results that $\theta_j^k \leq \theta_0^k$, so that we obtain the following relation from Lemma 2.3, by setting $M \geq \theta_0^k$:

$$\|\boldsymbol{x}^{(k+1)} - \hat{\boldsymbol{x}}\|^2 \leq M\|\boldsymbol{x}^{(0)} - \hat{\boldsymbol{x}}\|^2 - 2\alpha_{max}M\sum_{j=0}^{k}\lambda_j\nabla f(\boldsymbol{x}^{(j)})^T(\boldsymbol{y}^{(j)} - \boldsymbol{x}^{(j)}). \qquad (2.68)$$

Lemma 2.2 ensures that $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ is bounded, so that it has at least one limit point, denoted by $\boldsymbol{x}^\infty$. From Theorem 2.10, $\boldsymbol{x}^\infty$ is stationary; moreover, as f is convex, it is a minimum point, i.e.

$\boldsymbol{x}^\infty \in X^*$. Let $\{\boldsymbol{x}^{(k_i)}\}_{i\in\mathbb{N}}$ be a subsequence of $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ which converges to $\boldsymbol{x}^\infty$. By applying the same arguments employed to derive (2.68), for any fixed $i \in \mathbb{N}$ and for all $k \geq k_i$ we obtain

$$\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\infty\|^2 \leq M\|\boldsymbol{x}^{(k_i)} - \boldsymbol{x}^\infty\|^2 - 2\alpha_{max} M \sum_{j=k_i}^{k} \lambda_j \nabla f(\boldsymbol{x}^{(j)})^T(\boldsymbol{y}^{(j)} - \boldsymbol{x}^{(j)}). \tag{2.69}$$

Since $\{\boldsymbol{x}^{(k_i)}\}_{i\in\mathbb{N}}$ converges to $\boldsymbol{x}^\infty$ and $-\sum_{j=0}^{\infty} \lambda_j \nabla f(\boldsymbol{x}^{(j)})^T(\boldsymbol{y}^{(j)} - \boldsymbol{x}^{(j)})$ is a convergent series, for any $\varepsilon > 0$ there exists a sufficiently large integer k_i such that $\|\boldsymbol{x}^{(k_i)} - \boldsymbol{x}^\infty\|^2 \leq \varepsilon/2M$ and $-\sum_{j=k_i}^{k} \lambda_j \nabla f(\boldsymbol{x}^{(j)})^T(\boldsymbol{y}^{(j)} - \boldsymbol{x}^{(j)}) \leq \varepsilon/(4M\alpha_{max})$. Then, it follows from (2.69) that $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\infty\|^2 \leq \varepsilon$ for all $k \geq k_i$. Since ε can be chosen arbitrarily small, this means that the whole sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ converges to $\boldsymbol{x}^\infty$. □

The previous theorem gives an easily implementable rule to ensure the theoretical convergence of SGP to a solution.

In the following we report a result that ensures a $\mathcal{O}(1/k)$ convergence rate on the objective function value for the sequence of the iterates of the SGP scheme. This result is similar to the one reported in Theorem 2.1 for forward–backward methods with linesearch along the projection arc.

Theorem 2.12. *[19, Theorem 3.2] Assume that the hypotheses of Theorem 2.11 hold and, in addition, that assumption a) or b) of Proposition 2.3 is satisfied. Let f^* be the optimal function value for problem* (2.1)*. Then, we have*

$$f(\boldsymbol{x}^{(k)}) - f^* = \mathcal{O}(1/k).$$

Proof. If we define $\lambda_{\min}$ as in Proposition 2.3 and we set $a = 2\lambda_{\min}\alpha_{min}$, from (2.67) we have

$$\begin{aligned}\|\boldsymbol{x}^{(k+1)} - \hat{\boldsymbol{x}}\|^2_{D_k} \quad \leq \quad & \|\boldsymbol{x}^{(k)} - \hat{\boldsymbol{x}}\|^2_{D_k} - 2\alpha_k\lambda_k \nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}) + \\ & -2\lambda_k\alpha_k(f(\boldsymbol{x}^{(k)}) - f(\hat{\boldsymbol{x}}))\end{aligned}$$

As $\nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)})$ and $f(\hat{\boldsymbol{x}}) - f(\boldsymbol{x}^{(k)})$ are negative quantities, we have

$$\begin{aligned}&\|\boldsymbol{x}^{(k)} - \hat{\boldsymbol{x}}\|^2_{D_k} - 2\alpha_k\lambda_k \nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}) - 2\lambda_k\alpha_k(f(\boldsymbol{x}^{(k)}) - f(\hat{\boldsymbol{x}})) \\ &\leq \|\boldsymbol{x}^{(k)} - \hat{\boldsymbol{x}}\|^2_{D_k} - 2\alpha_{max}\lambda_k \nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}) + a(f(\hat{\boldsymbol{x}}) - f(\boldsymbol{x}^{(k)})),\end{aligned}$$

so that

$$\|\boldsymbol{x}^{(k+1)} - \hat{\boldsymbol{x}}\|^2_{D_k} \leq \|\boldsymbol{x}^{(k)} - \hat{\boldsymbol{x}}\|^2_{D_k} - 2\alpha_{max}\lambda_k \nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)} - \boldsymbol{x}^{(k)}) + a(f(\hat{\boldsymbol{x}}) - f(\boldsymbol{x}^{(k)})).$$

From inequality (2.61), we have that

$$\begin{aligned}
\frac{1}{\mu_k}\|\boldsymbol{x}^{(k+1)}-\hat{\boldsymbol{x}}\|^2 &\leq \|\boldsymbol{x}^{(k+1)}-\hat{\boldsymbol{x}}\|^2_{D_k}\\
&\leq \|\boldsymbol{x}^{(k)}-\hat{\boldsymbol{x}}\|^2_{D_k}-2\alpha_{max}\lambda_k\nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)}-\boldsymbol{x}^{(k)})+\\
&\quad +a(f(\hat{\boldsymbol{x}})-f(\boldsymbol{x}^{(k)}))\\
&\leq \mu_k\|\boldsymbol{x}^{(k)}-\hat{\boldsymbol{x}}\|^2-2\alpha_{max}\lambda_k\nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)}-\boldsymbol{x}^{(k)})+\\
&\quad +a(f(\hat{\boldsymbol{x}})-f(\boldsymbol{x}^{(k)})).
\end{aligned}$$

Thus, by multiplying the last inequality by μ_k we obtain

$$\begin{aligned}
\|\boldsymbol{x}^{(k+1)}-\hat{\boldsymbol{x}}\|^2 &\leq \mu_k^2\|\boldsymbol{x}^{(k)}-\hat{\boldsymbol{x}}\|^2-2\alpha_{max}\mu_k\lambda_k\nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)}-\boldsymbol{x}^{(k)})+\\
&\quad +\mu_k a(f(\hat{\boldsymbol{x}})-f(\boldsymbol{x}^{(k)})).
\end{aligned}$$

From the fact that $\mu_k \geq 1$ we have

$$\begin{aligned}
&\mu_k^2\|\boldsymbol{x}^{(k)}-\hat{\boldsymbol{x}}\|^2-2\alpha_{max}\mu_k\lambda_k\nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)}-\boldsymbol{x}^{(k)})+\mu_k a(f(\hat{\boldsymbol{x}})-f(\boldsymbol{x}^{(k)}))\\
&\leq \mu_k^2\|\boldsymbol{x}^{(k)}-\hat{\boldsymbol{x}}\|^2-2\alpha_{max}\mu_k^2\lambda_k\nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)}-\boldsymbol{x}^{(k)})+a(f(\hat{\boldsymbol{x}})-f(\boldsymbol{x}^{(k)})),
\end{aligned}$$

which yields

$$\|\boldsymbol{x}^{(k+1)}-\hat{\boldsymbol{x}}\|^2\leq \mu_k^2\|\boldsymbol{x}^{(k)}-\hat{\boldsymbol{x}}\|^2-2\alpha_{max}\mu_k^2\lambda_k\nabla f(\boldsymbol{x}^{(k)})^T(\boldsymbol{y}^{(k)}-\boldsymbol{x}^{(k)})+a(f(\hat{\boldsymbol{x}})-f(\boldsymbol{x}^{(k)})).$$

By repeatedly applying the last inequality we obtain

$$\begin{aligned}
\|\boldsymbol{x}^{(k+1)}-\hat{\boldsymbol{x}}\|^2 &\leq \theta_0^k\|\boldsymbol{x}^{(0)}-\hat{\boldsymbol{x}}\|^2-2\alpha_{max}\sum_{j=0}^{k}\theta_j^k\lambda_j\nabla f(\boldsymbol{x}^{(j)})^T(\boldsymbol{y}^{(j)}-\boldsymbol{x}^{(j)})+\\
&\quad +a((k+1)f(\hat{\boldsymbol{x}})-\sum_{j=0}^{k}f(\boldsymbol{x}^{(j)}))\\
&\leq M\|\boldsymbol{x}^{(0)}-\hat{\boldsymbol{x}}\|^2-2\alpha_{max}M\sum_{j=0}^{k}\lambda_j\nabla f(\boldsymbol{x}^{(j)})^T(\boldsymbol{y}^{(j)}-\boldsymbol{x}^{(j)})+\\
&\quad +a((k+1)f(\hat{\boldsymbol{x}})-\sum_{j=0}^{k}f(\boldsymbol{x}^{(j)})), \qquad (2.70)
\end{aligned}$$

where, as in the proof of Theorem 2.11, we set $\theta_j^k=\prod_{i=j}^k\mu_j^2$ and $M\geq\theta_j^k$. The Armijo rule in STEP 5 can be rewritten as

$$-\beta\lambda_k\nabla f(\boldsymbol{x}^{(k)})^T\boldsymbol{d}^{(k)}\leq f(\boldsymbol{x}^{(k)})-f(\boldsymbol{x}^{(k+1)}).$$

Summing the previous inequality for $k = 0, ..., j$ gives

$$\begin{aligned} -\beta \sum_{k=0}^{j} \lambda_k \nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{d}^{(k)} &\leq \textstyle\sum_{k=0}^{j} (f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^{(k+1)})) \\ &= f(\boldsymbol{x}^{(0)}) - f(\boldsymbol{x}^{(j+1)}). \end{aligned} \tag{2.71}$$

Thanks to inequality (2.71), we have

$$\begin{aligned} \|\boldsymbol{x}^{(k+1)} - \hat{\boldsymbol{x}}\|^2 \quad &\leq \quad M\|\boldsymbol{x}^{(0)} - \hat{\boldsymbol{x}}\|^2 + \frac{2\alpha_{max} M}{\beta}(f(\boldsymbol{x}^{(0)}) - f(\hat{\boldsymbol{x}})) + \\ &\quad +a(kf(\hat{\boldsymbol{x}}) - \sum_{j=1}^{k} f(\boldsymbol{x}^{(j)})), \end{aligned} \tag{2.72}$$

where we also added the positive quantity $a(f(\boldsymbol{x}^{(0)}) - f(\hat{\boldsymbol{x}}))$ to the right hand side of (2.70). Moreover, exploiting the inequality

$$0 \leq \sum_{j=0}^{k} j(f(\boldsymbol{x}^{(j)}) - f(\boldsymbol{x}^{(j+1)})) = \sum_{j=1}^{k} f(\boldsymbol{x}^{(j)}) - kf(\boldsymbol{x}^{(k+1)})$$

gives

$$\begin{aligned} \|\boldsymbol{x}^{(k+1)} - \hat{\boldsymbol{x}}\|^2 \quad &\leq \quad M\|\boldsymbol{x}^{(0)} - \hat{\boldsymbol{x}}\|^2 + \frac{2\alpha_{max} M}{\beta}(f(\boldsymbol{x}^{(0)}) - f(\hat{\boldsymbol{x}})) + \\ &\quad +ak(f(\hat{\boldsymbol{x}}) - f(\boldsymbol{x}^{(k+1)})). \end{aligned}$$

After suitable rearrangement of the terms, the desired thesis is proved:

$$f(\boldsymbol{x}^{(k+1)}) - f(\hat{\boldsymbol{x}}) \leq \frac{M}{ak}\left(\|\boldsymbol{x}^{(0)} - \hat{\boldsymbol{x}}\|^2 + 2\frac{\alpha_{max}}{\beta}(f(\boldsymbol{x}^{(0)}) - f(\hat{\boldsymbol{x}}))\right).$$

□

Scaling matrix choice

The most extensively exploited rule to specify the scaling matrix D_k is provided in [75, 76], when the solution of problem (2.1) is forced to be nonnegative in each component, namely when $\Omega = \{\boldsymbol{x} \in \mathbb{R}^n : \ \boldsymbol{x} \geq 0\}$. This technique is based on the decomposition of the gradient into two parts:

$$\nabla f(\boldsymbol{x}) = V(\boldsymbol{x}) - U(\boldsymbol{x}), \quad V(\boldsymbol{x}) > 0, \ U(\boldsymbol{x}) \geq 0. \tag{2.73}$$

It is worth noticing that this approach can be employed without efforts in the field of image reconstruction, in which the gradient of the objective function can naturally be decomposed in the form (2.73) for most of the adopted models.

If $\boldsymbol{x}^* \in \Omega$ is a solution of problem (2.1), then $\boldsymbol{x}^*$ must satisfy the Karush–Kuhn–Tucker (KKT) conditions

$$\nabla f(\boldsymbol{x}^*) - \lambda = 0, \quad \boldsymbol{x}^* \geq 0, \quad \lambda \geq 0, \quad \boldsymbol{x}_i^* \lambda_i = 0, \quad i = 1, \dots, n \tag{2.74}$$

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

$$\boldsymbol{x}_i^* \nabla f(\boldsymbol{x}_i^*) = 0, \quad i = 1, \dots, n. \tag{2.75}$$

On the basis of the decomposition (2.73), the n nonlinear equations (2.75) can also be rewritten as the vectorial fixed point equation

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

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

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

which is equivalent to

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

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

$$D_k^{-1} = \operatorname{diag}\left(\frac{\boldsymbol{x}_1^{(k)}}{V_1(\boldsymbol{x}^{(k)})}, \dots, \frac{\boldsymbol{x}_n^{(k)}}{V_n(\boldsymbol{x}^{(k)})} \right). \tag{2.76}$$

As a consequence, in case of nonnegativity constraints it comes quite natural to address problem (2.1) by means of a scaled gradient method with steplength equal to 1. Thus, the idea proposed in [20] and subsequent works is to adopt the matrix (2.76) into Algorithm 9, with the further request of forcing its eigenvalues to belong to the bounded interval $[1/\mu_k, \mu_k]$, in order to comply with the convergence assumptions of Theorem 2.11:

$$(D_k^{-1})_{ii} = \max\left\{ \min\left\{ \frac{\boldsymbol{x}_i^{(k)}}{V_i(\boldsymbol{x}^{(k)})}, \mu_k \right\}, \frac{1}{\mu_k} \right\}, \quad i = 1, \dots, n. \tag{2.77}$$

As well as the choice suggested for Algorithm 8, the matrix D_k^{-1} 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

We report in this paragraph some considerations concerning the most exploited choices for the steplength parameter α_k. Due to the large success of the Barzilai–Borwein rules (2.20) in the context of unconstrained optimization, it came natural to extend the BB–like schemes described in Section 2.2 to the SGP method. A first extension was conceived for gradient projection methods in [14], where two GP schemes denominated Spectral Projected Gradient (SPG) methods were proposed: both schemes were equipped with the choice $\alpha_k = \alpha_k^{BB1}$ for the steplength, the first one performing the linesearch on λ_k along the arc and the second one along the feasible direction. Later, the theory is extended to scaled gradient projection methods in [15], while the practical numerical experiments considered only the non–scaled scheme. A scaling matrix was first introduced into the determination of the two BB rules in [20], by imposing the secant equations (2.18)–(2.19) to the matrix $B(\alpha_k) = (\alpha_k D_k^{-1})^{-1}$, which yielded to the following rules

$$\alpha_k^{BB1S} = \frac{s^{(k-1)^T} D_k D_k s^{(k-1)}}{s^{(k-1)^T} D_k \boldsymbol{y}^{(k-1)}} \quad ; \quad \alpha_k^{BB2S} = \frac{s^{(k-1)^T} D_k^{-1} \boldsymbol{y}^{(k-1)}}{\boldsymbol{y}^{(k-1)^T} D_k^{-1} D_k^{-1} \boldsymbol{y}^{(k-1)}}. \tag{2.78}$$

Furthermore, inspired by the alternation strategy (2.41) implemented in the framework of non–scaled gradient methods, in [20] the authors proposed a steplength updating rule for SGP which adaptively alternates the values provided in (2.78), as detailed in Algorithm 10.

Algorithm 10 is a modification of rule (2.41), where the alternation of the two steplengths is no more determined by a constant parameter τ as in (2.41), but with a variable threshold τ_k. Thus, the choice of τ_0 become less important for the SGP performance and, in the authors' experience, it seems able to avoid the drawbacks due to the use of the same steplength rule in too many consecutive iterations.

Successively, the limited–memory steplength rule devised in [54] and based on the Ritz–like values of the tridiagonal matrix (2.48) was transposed into the SGP framework when Ω is the nonnegativity constraint set, as suggested in [91]. 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 (2.43) by taking into account the presence of a scaling matrix and the projection onto the feasible set. The rule was devised exploiting the fact that applying a scaled gradient method $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k D_k^{-1} \nabla f(\boldsymbol{x}^{(k)})$, with D_k symmetric and positive definite, to the minimization of a function f is equivalent to performing the change of variables $\boldsymbol{x} = D_k^{-1/2}\boldsymbol{y}$ and addressing the following scaled problem

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

by means of a steepest descent method

$$\boldsymbol{y}^{(k+1)} = \boldsymbol{y}^{(k)} - \alpha_k \nabla \widetilde{f}(\boldsymbol{y}^{(k)}) \tag{2.79}$$

Algorithm 10 Steplength Selection 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 $s^{(k-1)^T} D_k \boldsymbol{y}^{(k-1)} \leq 0$ THEN
        $\alpha_k^{(1)} = \alpha_{max}$;
    ELSE
        $\alpha_k^{(1)} = \min\left\{\alpha_{max}, \max\{\alpha_{min}, \alpha_k^{BB1S}\}\right\}$;
    ENDIF

    IF $s^{(k-1)^T} D_k^{-1} \boldsymbol{y}^{(k-1)} \leq 0$ 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
        $\alpha_k = \min\left\{\alpha_j^{(2)},\ j = \max\{1, k - M_\alpha\}, \dots, k\right\}$;    $\tau_{k+1} = \tau_k \cdot 0.9$;
    ELSE
        $\alpha_k = \alpha_k^{(1)}$;    $\tau_{k+1} = \tau_k \cdot 1.1$.
    ENDIF

ENDIF
```

with respect to the variable $\boldsymbol{y}$ [12]. The previous remark led to the idea of applying the limited-memory scheme to the method (2.79) instead of the scaled version of it and also, as $\nabla\widetilde{f}(\boldsymbol{y}^{(k)}) = D_k^{-1/2}\nabla f(\boldsymbol{x}^{(k)})$, to store at each iteration the scaled gradient $D_k^{-1/2}\boldsymbol{g}^{(k)}$ instead of $\boldsymbol{g}^{(k)}$. Furthermore, the nonnegativity constraint was addressed by looking at the complementarity condition (2.75) satisfied by the solution of problem (2.1), for which the components of the gradient related to inactive constraints in the solution need 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

$$\widetilde{g}_j^{(k)} = \begin{cases} 0 & \text{if } \boldsymbol{x}_j^{(k)} = 0, \\ \left(\nabla f(\boldsymbol{x}^{(k)})\right)_j & \text{if } \boldsymbol{x}_j^{(k)} > 0. \end{cases} \tag{2.80}$$

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

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

The subsequent m Ritz–like values θ_i, $i = 1, \dots, m$, are then computed by following the same passages included in equations (2.46)–(2.48) with G and $\boldsymbol{g}^{(k)}$ replaced by $\widetilde{G}$ and $D_k^{-1/2} \widetilde{\boldsymbol{g}}^{(k)}$. It is worth noticing that, for small m, this generalized limited–memory approach is not much more expensive than the BB–like schemes previously described. Indeed, if we assume that D_k is diagonal, each sweep requires

- the computation of m scaled gradients $D_j^{-1/2} \widetilde{\boldsymbol{g}}^{(j)}$ 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 $R^T r = \widetilde{G}^T D_k^{-1/2} \widetilde{\boldsymbol{g}}^{(k)}$, 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 (2.78) for m iterations requires $3m$ vector–vector products. Therefore, if we assume, for example, to choose $m = 3$, the limited–memory approach has a computational cost of $\mathcal{O}(9n)$ products as well as the two BB steplengths.

Chapter 3

Application to Spherical Deconvolution in diffusion MRI

The problem of detecting and tracking fibre paths on the brain is one of the major challenges in diffusion Magnetic Resonance Imaging (dMRI): in this framework, the random movement of molecules in the white matter is exploited to gather informations on fibre orientations and brain connections. This feature is particulary effective for medical applications as it allows to investigate the structures of the nervous system *in vivo* and noninvasively.

Several High Angular Resolution Diffusion Imaging (HARDI) approaches have been proposed to provide accurate estimation of fibre populations in a time suitable for clinical application. Many investigation methods of this area are based on Spherical Deconvolution (SD), which models the signal attenuation acquired with dMRI as a convolution between a given response function and the fibre orientation distribution.

The variable metric methods presented in the previous section are here exploited to solve the optimization problems deriving from signal reconstruction into this field of application. In Section 3.1 we state the applicative problem, while in Section 3.2 we detail the settings of the forward–backward methods that we employed to tackle it; finally, in Section 3.3 we report some results of the numerical experience.

3.1 Problem formulation

In this section we recall one of the theoretical approaches aimed to recover fibre orientations in the white matter of the brain. Spherical deconvolution methods [1, 47, 113, 114] rely on the assumptions that Diffusion Weighted (DW) signals can be expressed as a convolution of a single fibre response function (RF) with the fibre orientation distribution (FOD). The FOD is a function on the unit sphere which models the direction and volume fractions of fibres in a voxel; the RF corresponds to the DW signal of a single fibre compartment. DW signals can be modeled by the Gaussian Mixture Model [2, 35] for N fibers:

$$\frac{S(s)}{S_0} = \sum_{j=1}^{N} v_j exp\Big[-\hat{b}\big(\beta_j + \alpha_j \langle s, r_j \rangle^2\big)\Big] \tag{3.1}$$

where

- S is the DW signal normalized by the non–diffusion weighted MRI signal S_0;
- s is the unit vector in the direction of the diffusion gradient;
- for each fibre $j = 1, \dots, N$, r_j is a unit vector along the direction of the j–th fibre and pointing in a given hemisphere;
- v_j is the partial volume corresponding to the j–th fibre compartment;
- $\hat{b}$ is the diffusion weighting factor;
- β_j corresponds to the radial diffusivities and α_j to the difference between the longitudinal diffusivity and β_j.

The FOD f can expressed as a linear combination of basis functions [93]

$$f(r) = \sum_{j=1}^{N} v_j \delta(r + r_j) \tag{3.2}$$

where δ denotes the Dirac delta function on the unit sphere $\mathbb{S}^2$; thus, formulation (3.1) can be written as [70]

$$\frac{S(s)}{S_0} = \int_{\mathbb{S}^2} H\big(\langle s, r \rangle\big) f(r) \mu(dr) \tag{3.3}$$

where $H(\langle s, r \rangle) = \exp(-\hat{b}(\beta + \alpha \langle s, r \rangle^2))$, $\mu(dr)$ is the standard measure on the unit sphere and $\langle \cdot, \cdot \rangle$ denotes the Euclidean scalar product.

The problem of recovering the FOD estimation function (3.3) can then be expressed in discretized linear form [47, 113]

$$\boldsymbol{b} = \Phi \boldsymbol{x} + \boldsymbol{\eta} \tag{3.4}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the vector of the FOD coefficients, $\boldsymbol{b} \in \mathbb{R}^m$ is the vector of the dMRI measurements; $\Phi = (\Phi_{i,j})$ is a rectangular matrix given by

$$\Phi_{i,j} = \exp(-\hat{b}(\beta + \alpha \langle s_i, u_j \rangle^2))$$

which models the convolution operator, with u_j are the unit vectors associated with the FOD sampling points on the unit sphere; $\boldsymbol{\eta}$ is the acquisition Gaussian noise.

The deconvolution problem (3.4) is intrinsically ill–posed and it is necessary to apply some regularization schemes to recover a unique solution for the problem. The small number of fibre directions in each voxel that corresponds to the FOD coefficients x_i suggests some inherent sparsity for problem (3.4) and Compressive Sampling (CS) theory [23] can be exploited by using sparsity priors as regularizers. Many methods proposed for the solution of problem (3.4) exploit sparse regularization by means of ℓ_1 minimisation [70, 81, 93] and in [36] the authors propose ℓ_0-based approach performing the reconstruction on a voxel–by–voxel level. Recently, a quantitative comparison was proposed between the latter method and the approach [82] based on the well–known Iterative Image Space Reconstruction Algorithm [42]. The ℓ_0-based work was extended in [3] where the fibre configuration is solved on all voxels of interest simultaneously, aiming at taking into account both voxelwise sparsity and the spatial coherence of the fibre orientation between neighbour voxels. The approaches developed in [36] and [3] produce a large–scale problem scheme suitable to be tackled by first order methods and an interesting framework to test variable metrics acceleration techniques introduced in Section 2.4. The general configuration of the aforementioned ℓ_1 minimization procedure is recalled in the following.

According to [36], problem (3.4) considered for a single voxel v on a domain of Λ total voxels can be reformulated as a constrained ℓ_0 minimization problem

$$\min_{\boldsymbol{x}_v \in \Omega} \ \tfrac{1}{2} \left\| \Phi_{mn} \boldsymbol{x}_v - \boldsymbol{b}_v \right\|_2^2 , \qquad \Omega = \left\{ \boldsymbol{x}_v \geq \boldsymbol{0}, \quad \| \boldsymbol{x}_v \|_0 \leq \kappa \right\} , \tag{3.5}$$

where Φ_{mn} is an $m \times n$ sensing matrix, $\boldsymbol{x}_v \in \mathbb{R}^n$ represents the FOD in the voxel indexed v, $\boldsymbol{b}^{(v)} \in \mathbb{R}^m$ is the acquired signal corresponding to voxel v, κ is the expected maximum quantity of fibre populations in v and $\|\cdot\|_0$ represents the ℓ_0 norm of a vector, i.e., the number of nonzero components of the vector, which is a non convex function. A reweighted ℓ_1 minimization scheme was first introduced in [25] with the aim to tackle ℓ_0 minimisation by a sequence of convex weighted ℓ_1 problems of the form

$$\min_{\boldsymbol{x}_v^{(t)} \in \Omega} \ \tfrac{1}{2} \left\| \Phi_{mn} \boldsymbol{x}_v^{(t)} - \boldsymbol{b}_v \right\|_2^2 , \qquad \Omega = \left\{ \boldsymbol{x}_v^{(t)} \geq \boldsymbol{0}, \quad \left\| \boldsymbol{x}_v^{(t)} \right\|_{\boldsymbol{w}^{(t)},1} \leq \kappa \right\} , \tag{3.6}$$

where the weighted ℓ_1 norm is defined by $\|\boldsymbol{x}\|_{\boldsymbol{w},1} = \sum_j w_j |x_j|$. At each iteration t of the sequence, the weights are assigned as an approximation of the inverse values of the solution $w_j^{(t+1)} \approx 1/(x_j^{(t)} + \epsilon)$, $\epsilon > 0$. The sequence of weighted ℓ_1 problems aims at approximating the ℓ_0 problem at convergence. The approaches described so far can be efficiently solved by means of the LARS algorithm [49], whose implementation is available in the open–source toolbox named SPArse Modeling Software (SPAMS) [80]. The idea of exploiting the anatomical coherence of fibre tracts led to the extension of this method to the whole volume of the brain. According to [3], if Λ voxels are considered, by concatenating the vectors $\boldsymbol{x}_v$ columnwise a vector $\boldsymbol{X} \in \mathbb{R}^N$ can be built, whose columns correspond to the FODs of each voxel ($N = n \times \Lambda$). Signal vectors $\boldsymbol{b}_v$ are concatenated as well producing a vector $\boldsymbol{B} \in \mathbb{R}^M$ with $M = m \times \Lambda$ and an $M \times N$

sparse sensing matrix Φ is obtained by repetition of Λ blocks of the matrix Φ_{mn} such that $\Phi\boldsymbol{X} = [\Phi_{mn}\boldsymbol{x}_{(v)}]_{v\in 1,\dots,\Lambda} \in \mathbb{R}^M$. Formulation (3.6) is then solved simultaneously in the entire brain volume, leading to the following sequence of constrained large–scale problems

$$\min_{\boldsymbol{X}^{(t)}\in\Omega} \quad F(\boldsymbol{X}^{(t)}) = \tfrac{1}{2}\left\|\Phi\boldsymbol{X}^{(t)} - \boldsymbol{B}\right\|_2^2, \qquad \Omega = \left\{\boldsymbol{X}^{(t)} \geq \boldsymbol{0}, \quad \left\|\boldsymbol{X}^{(t)}\right\|_{\boldsymbol{W}^{(t)},1} \leq K\right\}, \tag{3.7}$$

where $\left\|\boldsymbol{X}\right\|_{\boldsymbol{W},1} = \sum_{i=1}^{N} W_i X_i$ and $K = \kappa\Lambda$ is the estimated maximum number of fibres to be detected in the whole brain. The weight vectors $\boldsymbol{W}^{(t)}$ are computed at each step t with the aim to exploit spatial and angular coherence of fibre bundles, i.e. the idea that fibres in "neighbour" voxel should take similar directions [3].

Problems (3.7) can be seen as special instances of problem (2.1) considered in Chapter 2, where $\Omega = \left\{\boldsymbol{X}^{(t)} \geq \boldsymbol{0}, \quad \left\|\boldsymbol{X}^{(t)}\right\|_{\boldsymbol{W}^{(t)},1} \leq K\right\}$ is a closed, convex and nonempty subset of $\mathbb{R}^N$ and the function $f(\boldsymbol{X}) = \frac{1}{2}\left\|\Phi\boldsymbol{X}^{(t)} - \boldsymbol{B}\right\|_2^2$ is the least squares fit–to–data functional.

3.2 Optimization methods

In this section we will describe the implementation details designed for Algorithm 8, Section 2.4.1 (denoted in the following as Scaled GP_Ex) and SGP (Section 2.4.2, Algorithm 9) methods employed to solve the sequence of problems (3.7). For sake of clarity, we report here the general iteration of Scaled GP_Ex algorithm

$$\begin{aligned} \boldsymbol{y}^{(k)} &= \boldsymbol{x}^{(k)} + \beta_k(\boldsymbol{x}^{(k)} - x^{(k-1)}) \\ \boldsymbol{x}^{(k+1)} &= P_{\Omega,D_k}(\boldsymbol{y}^{(k)} - \alpha_k D_k^{-1}\nabla f_0(\boldsymbol{y}^{(k)})) \end{aligned} \tag{3.8}$$

and of SGP algorithm:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \lambda_k\left(P_{\Omega,D_k}(\boldsymbol{x}^{(k)} - \alpha_k D_k^{-1}\nabla f(\boldsymbol{x}^{(k)})) - \boldsymbol{x}^{(k)}\right). \tag{3.9}$$

3.2.1 Scaling matrix and steplength selection

The variable metric strategies implemented in the numerical tests are reported in the following.

Scaling matrix

We denote as $\mathcal{D}_\rho$ the set of symmetric positive definite matrices D with eigenvalues τ_j such that

$$0 < \frac{1}{\rho} \leq \tau_j \leq \rho, \;\; j = 1, \dots, N$$

and $P_{\Omega,D}(\boldsymbol{v}) = \text{argmin}_{\boldsymbol{u}\in\Omega}(\boldsymbol{u} - \boldsymbol{v})^T D(\boldsymbol{u} - \boldsymbol{v})$.
For both the considered methods, the scaling matrices $D_i \in \mathcal{D}_\rho$ are chosen as diagonal matrices

in order to avoid significant computational costs: so, for each iteration i, $D_i = (D_i)_{j,j} =: \tau_j^{(i)}$ for $j = 1, \dots, N$.

As already pointed out in Section 2.4, the assumptions of Theorems 2.8, 2.9, 2.11 and 2.12 are fulfilled when, for any i, the eigenvalues $\tau_j^{(i)}$ of D_i are such that

$$0 < \frac{1}{\rho_i} \leq \tau_j^{(i)} \leq \rho_i,\ j = 1, \dots, N\ ,\ \ \rho_i^2 = 1 + \theta_i\ ,\ \ \sum_{i=0}^{\infty} \theta_i < \infty.$$

For the convergence of the iterates of Scaled GP_Ex, an additional requirement for the values of θ_i is also needed

$$\{\theta_i\} = \mathcal{O}\Big(\frac{1}{i^p} \Big)\ ,\ p > 2.$$

Thus, following the technique described in Section 2.4.2, we decompose the gradient of the function F as

$$\nabla F(\boldsymbol{X}) = V(\boldsymbol{X}) - U(\boldsymbol{X})\quad ,\quad V(\boldsymbol{X}) > \mathbf{0}\ ,\ U(\boldsymbol{X}) \geq \mathbf{0}$$

by choosing

$$V(\boldsymbol{X}) = \Phi^T \Phi \boldsymbol{X}\quad ,\quad U(\boldsymbol{X}) = \Phi^T \boldsymbol{B}. \tag{3.10}$$

As a consequence, we equip the SGP and the Scaled GP_Ex algorithms with the following scaling strategy:

$$(D_i)_{j,j} = \max\left\{ \frac{1}{\rho_i}, \min\left\{ \rho_i, \frac{(\boldsymbol{z}^{(i)})_j}{(\Phi^T \Phi \boldsymbol{z}^{(i)})_j} \right\} \right\}, \quad j = 1, \dots, N, \qquad \rho_i = \sqrt{1 + \frac{\gamma}{i^p}}.$$

where $\boldsymbol{z}^{(i)} = \boldsymbol{x}^{(i)}$ for the SGP method, $\boldsymbol{z}^{(i)} = \boldsymbol{y}^{(i)}$ for the Scaled GP_Ex method, $\gamma > 0$ and $p > 2$. This updating rule satisfy the convergence conditions and doesn't add remarkable computational cost since the vector $\Phi^T \Phi \boldsymbol{z}^{(i)}$ is available from the gradient.
In the numerical experiments of the next section, we set $\gamma = 10^{13}$ and $p = 2.1$.

Steplength selection

For each iteration i, the steplength α_i for Scaled GP_Ex is constant or eventually computed via a backtracking algorithm, while the extrapolation parameter β_i is chosen accordingly to the following rule in order to ensure the convergence of the method:

$$\beta_i = \frac{i-1}{i+2.1}.$$

The steplength rule introduced in [91] and described in Section 2.4.2 (formulae (2.79) - (2.80)) is here employed with $m = 3$

$$\alpha_{i+j} = \frac{1}{\theta_j},\ j = 1, 2, 3.$$

for the determination of SGP steplengths.

3.2.2 Computation of the projection

The set of feasible points for the solution of problem (3.7) is defined by nonnegativity and linear inequality constraints; thus, the problem of projecting a point $\boldsymbol{z} \in \mathbb{R}^n$ onto Ω can be seen as a particular case of projecting onto a set $\tilde{\Omega}$

$$\tilde{\Omega} = \left\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{a} \leq \boldsymbol{x} \leq \boldsymbol{b}, \quad \|\boldsymbol{x}\|_{\boldsymbol{W},\mathbf{1}} \leq c\right\} = \left\{\boldsymbol{x} \in \mathbb{R}^n \quad | \quad \boldsymbol{a} \leq \boldsymbol{x} \leq \boldsymbol{b}, \quad \boldsymbol{W}^T\boldsymbol{x} \leq c, \boldsymbol{W} \in \mathbb{R}^n\right\}, \tag{3.11}$$

defined by box and linear inequality constraints, whose definition is detailed in the following.

Given a nonempty closed convex set $S \subset \mathbb{R}^n$ and a vector $\boldsymbol{z} \in \mathbb{R}^n$, we are interested in performing the projection in the norm induced by a diagonal symmetric and positive definite matrix D

$$\|\boldsymbol{x}\|_D = \sqrt{\boldsymbol{x}^T D\boldsymbol{x}}, \qquad D = \operatorname{diag}(\boldsymbol{d}) = \operatorname{diag}(d_1, \ldots, d_n), \quad d_i > 0$$

that is, we need to find

$$\boldsymbol{x}^* = \mathrm{P}_{S,D}(\boldsymbol{z}) = \arg\min_{\boldsymbol{x}\in S} \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_D^2 = \arg\min_{\boldsymbol{x}\in S} \tilde{f}(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T D\boldsymbol{x} - \boldsymbol{z}^T D\boldsymbol{x} \tag{3.12}$$

We now show that the solution of problem (3.12) for $S = \tilde{\Omega}$ can be found by means of Algorithm 11, where

$$\Omega_{eq} = \{\boldsymbol{x} \in \mathbb{R}^n \quad | \quad \boldsymbol{a} \leq \boldsymbol{x} \leq \boldsymbol{b}, \quad \textstyle\sum_{i=1}^n W_i x_i = c\}.$$

Algorithm 11 Projection onto a region defined by a linear inequality and box constraints

```
Given a point $\boldsymbol{z} \in \mathbb{R}^n$, set the vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$, $\boldsymbol{a} \leq \boldsymbol{b}$, $\boldsymbol{W} \in \mathbb{R}^n$ and the constant $c \in \mathbb{R}$.
$\boldsymbol{x}_{box} = \max\{\boldsymbol{a}, \min\{\boldsymbol{b}, \boldsymbol{z}\}\}$
IF $\sum_{i=1}^n W_i(\boldsymbol{x}_{box})_i \leq c$
    $\boldsymbol{x}^* = \boldsymbol{x}_{box}$
ELSE
    $\boldsymbol{x}^* = \mathrm{P}_{\Omega_{eq},D}(\boldsymbol{z})$
```

We first recall the Karush-Kuhn-Tucker (KKT) conditions for the quadratic programming problem arising from projecting on

$$\Omega_{eq} = \{\boldsymbol{x} \in \mathbb{R}^n \quad | \quad \boldsymbol{a} \leq \boldsymbol{x} \leq \boldsymbol{b}, \quad \textstyle\sum_{i=1}^n W_i x_i = c\}$$

If $\hat{\boldsymbol{x}}$ is a local minimizer for problem (3.12) with $S = \Omega_{eq}$, then there exist Lagrange multipliers

$\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}} \in \mathbb{R}^n$ and $\hat{\lambda} \in \mathbb{R}$ such that

$$\nabla_x \mathcal{L}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}, \hat{\lambda}) = D\hat{\boldsymbol{x}} - D\boldsymbol{z} + \hat{\lambda}\boldsymbol{W} - \hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\beta}} = \boldsymbol{0}, \tag{3.13a}$$

$$c - \sum_{i=1}^{n} W_i \hat{x}_i \geq 0, \tag{3.13b}$$

$$\hat{\boldsymbol{x}} \geq \boldsymbol{a}, \quad -\hat{\boldsymbol{x}} \geq -\boldsymbol{b}, \tag{3.13c}$$

$$\hat{\boldsymbol{\alpha}} \geq \boldsymbol{0}, \quad \hat{\boldsymbol{\beta}} \geq \boldsymbol{0}, \tag{3.13d}$$

$$\boldsymbol{\alpha}^T(\hat{\boldsymbol{x}} - \boldsymbol{a}) = 0, \tag{3.13e}$$

$$\boldsymbol{\beta}^T(\boldsymbol{b} - \hat{\boldsymbol{x}}) = 0. \tag{3.13f}$$

We now write down the KKT conditions for problem (3.12) with $S = \tilde{\Omega}$; if $\boldsymbol{x}^*$ is a local minimizer, then there exist Lagrange multipliers $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$ such that

$$\nabla_x \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda) = D\boldsymbol{x}^* - D\boldsymbol{z} + \lambda\boldsymbol{W} - \boldsymbol{\alpha} + \boldsymbol{\beta} = \boldsymbol{0}, \tag{3.14a}$$

$$c - \sum_{i=1}^{n} W_i x_i^* \geq 0, \tag{3.14b}$$

$$\boldsymbol{x}^* \geq \boldsymbol{a}, \quad -\boldsymbol{x}^* \geq -\boldsymbol{b}, \tag{3.14c}$$

$$\lambda \geq 0, \quad \boldsymbol{\alpha} \geq \boldsymbol{0}, \quad \boldsymbol{\beta} \geq \boldsymbol{0} \tag{3.14d}$$

$$\boldsymbol{\alpha}^T(\boldsymbol{x}^* - \boldsymbol{a}) = 0, \quad \boldsymbol{\beta}^T(\boldsymbol{b} - \boldsymbol{x}^*) = 0, \tag{3.14e}$$

$$(c - \sum_{i=1}^{n} W_i x_i^*)\lambda = 0. \tag{3.14f}$$

From (3.14a) we have that

$$\boldsymbol{\alpha} - \boldsymbol{\beta} = D(\boldsymbol{x}^* - \boldsymbol{z}) + \lambda\boldsymbol{W} \equiv \nabla\tilde{f}(\boldsymbol{x}^*) + \lambda\boldsymbol{W}. \tag{3.15}$$

Moreover, if we denote $\mathcal{I} = \{i \mid a_i < x_i^* < b_i\}$ the set of inactive components' indexes of the vector $\boldsymbol{x}^*$ and with $\mathcal{I}_a = \{i \mid x_i^* = a_i\}$ and $\mathcal{I}_b = \{i \mid x_i^* = b_i\}$ the sets of actives' ones, from (3.14d) and (3.15) we have

$$\alpha_i = 0, \quad \beta_i = 0, \quad -[\nabla\tilde{f}(\boldsymbol{x}^*)]_i = \lambda W_i, \quad i \in \mathcal{I},$$

that is, the i-th component of the gradient is equal to the quantity $-\lambda W_i$ for all $i \in \mathcal{I}$.

We define the following indexes' sets with respect to the point $\boldsymbol{z}$

$$\mathcal{Z} = \{i \mid a_i < z_i < b_i\}, \qquad \mathcal{Z}_a = \{i \mid z_i \leq a_i\}, \quad \mathcal{Z}_b = \{i \mid z_i \geq b_i\}$$

We set $\boldsymbol{x}_{box} := \max\{\boldsymbol{a}, \min\{\boldsymbol{b}, \boldsymbol{z}\}\}$ and we first suppose that $\sum_{i \in \mathcal{Z}} W_i(\boldsymbol{x}_{box})_i \leq c$. By setting $\boldsymbol{x}^* = \boldsymbol{x}_{box}$, we can choose $\lambda^* = 0$ and we have that $\alpha_j^* = 0$, $\beta_j^* = 0$ for $j \in \mathcal{Z}$. Moreover,

$\alpha_i^* = D_{i,i}(a_i - z_i)$ and $\beta_i^* = 0$ for $i \in \mathcal{Z}_a$ and $\alpha_i^* = 0$, $\beta_i^* = D_{i,i}(z_i - b_i)$ for $i \in \mathcal{Z}_b$. With these choices for $(\boldsymbol{x}^*, \lambda^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ the KKT conditions (3.14) are satisfied:

$$\begin{aligned}
[D(\boldsymbol{x^*} - \boldsymbol{z}) + \lambda^* \boldsymbol{W} - \boldsymbol{\alpha}^* + \boldsymbol{\beta}^*]_j &= D_{j,j}(z_j - z_j) = 0,\ j \in \mathcal{Z} \\
[\nabla_x \mathcal{L}(\boldsymbol{x}^*, \lambda^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)]_i &= D_{i,i}(a_i - z_i) - D_{i,i}(a_i - z_i) = 0,\ i \in \mathcal{Z}_a, \qquad (3.16a) \\
[\nabla_x \mathcal{L}(\boldsymbol{x}^*, \lambda^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)]_i &= D_{i,i}(b_i - z_i) + D_{i,i}(z_i - b_i) = 0,\ i \in \mathcal{Z}_b \\
c - \textstyle\sum_{i=1}^n W_i x_i^* &= c - \textstyle\sum W_i (\boldsymbol{x}_{box})_i \geq 0 \qquad (3.16b) \\
\boldsymbol{x}^* \geq \boldsymbol{a}, \qquad &-\boldsymbol{x}^* \geq -\boldsymbol{b} \qquad (3.16c) \\
\lambda^* \geq 0, \qquad \boldsymbol{\alpha}^* \geq \boldsymbol{0}, \qquad &\boldsymbol{\beta}^* \geq \boldsymbol{0}, \qquad (3.16d) \\
\boldsymbol{\alpha}^{*T}(\boldsymbol{x}^* - \boldsymbol{a}) = \textstyle\sum \boldsymbol{\alpha}_i^*(x_i^* - a_i) = 0, \quad \boldsymbol{\beta}^{*T}(\boldsymbol{b} - \boldsymbol{x}^*) &= \textstyle\sum \boldsymbol{\beta}_i^*(b_i - x_i^*) = 0 \qquad (3.16e) \\
(c - \textstyle\sum_{i=1}^n W_i x_i^*)\, \lambda^* &= 0 \qquad (3.16f)
\end{aligned}$$

We now suppose that $\sum W_i (\boldsymbol{x}_{box})_i > c$ and we set $\boldsymbol{x}^* = \hat{\boldsymbol{x}} = \mathrm{P}_{\Omega_{eq},D}(\boldsymbol{z})$, $\lambda^* = \hat{\lambda}$, $\boldsymbol{\alpha}^* = \hat{\boldsymbol{\alpha}}$, $\boldsymbol{\beta}^* = \hat{\boldsymbol{\beta}}$. From KKT equations (3.13) we derive

$$\begin{aligned}
\nabla_x \mathcal{L}\left(\boldsymbol{x}^*, \lambda^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*\right) &= \boldsymbol{0} \qquad (3.17a) \\
c - \textstyle\sum_{i=1}^n W_i x_i^* &= 0 \\
\boldsymbol{x}^* \geq \boldsymbol{a}, \qquad -\boldsymbol{x}^* &\geq -\boldsymbol{b} \\
\boldsymbol{\alpha}^* \geq \boldsymbol{0}, \qquad \boldsymbol{\beta}^* &\geq \boldsymbol{0} \\
\boldsymbol{\alpha}^{*T}(\boldsymbol{x}^* - \boldsymbol{a}) = 0, \qquad \boldsymbol{\beta}^{*T}(\boldsymbol{b} - \boldsymbol{x}^*) &= 0.
\end{aligned}$$

We now must prove that $\lambda^* \geq 0$. From (3.17a) we have $\lambda^* \boldsymbol{W} = D(\boldsymbol{z} - \boldsymbol{x}^*) + \boldsymbol{\alpha}^* - \boldsymbol{\beta}^*$. We observe that $\sum W_i (\boldsymbol{x}_{box})_i > c = \sum_{i=1}^n W_i x_i^*$, so that $\sum W_i \left[(\boldsymbol{x}_{box})_i - x_i^*\right] > 0$. As a consequence, $\exists\ j \mid W_j \left[(\boldsymbol{x}_{box})_j - x_j^*\right] > 0$, or, equivalently $\exists\ j \mid W_j \left[\max\{a_j, \min\{b_j, z_j\}\} - x_j^*\right] > 0$. Now we may have four different scenarios.

I. $a_j < x_j^* < b_j$ and $W_j > 0$

We have that $\max\{a_j, \min\{b_j, z_j\}\} - x_j^* > 0$ so that $a_j < x_j^* < \max\{a_j, \min\{b_j, z_j\}\} \leq z_j$. From $\alpha_j^* = 0$, $\beta_j^* = 0$ it follows $\lambda^* = \frac{1}{W_j} D_{j,j}\left(z_j - x_j^*\right) > 0$.

II. $a_j < x_j^* < b_j$ and $W_j < 0$

We have that $\max\{a_j, \min\{b_j, z_j\}\} - x_j^* < 0$ so that $b_j > x_j^* > \max\{a_j, \min\{b_j, z_j\}\} \geq z_j$. From $\alpha_j^* = 0$, $\beta_j^* = 0$ it follows $\lambda^* = \frac{1}{W_j} D_{j,j}\left(z_j - x_j^*\right) > 0$.

III. $x_j^* = a_j$.

We have that $\alpha_j^* \geq 0$, $\beta_j^* = 0$. Since $\max\{a_j, \min\{b_j, z_j\}\} > x_j^* = a_j$, we have that

$\max\{a_j, \min\{b_j, z_j\}\} \leq z_j$; furthermore, $W_j > 0$. It follows that $\lambda^* = \frac{1}{W_j}\left[\alpha_j + D_{j,j}\left(z_j - x_j^*\right)\right] > 0$.

IV. $x_j^* = b_j$.
We have that $\alpha_j^* = 0$, $\beta_j^* \geq 0$. Since $\max\{a_j, \min\{b_j, z_j\}\} < x_j^* = b_j$, we have that $\max\{a_j, \min\{b_j, z_j\}\} \geq z_j$; furthermore, $W_j < 0$. It follows that $\lambda^* = \frac{1}{W_j}\left[D_{j,j}\left(z_j - x_j^*\right) - \beta_j^*\right] > 0$.

Thus, the problem of projecting a point onto the set $\tilde{\Omega}$ is reduced to the problem of projecting onto the set Ω_{eq}, which can be addressed by several linear time algorithms available in the literature (see e.g. [39]).

3.3 Numerical experiments

We performed our experiments with the aim to evaluate effectiveness of the algorithms equipped with the acceleration strategies described in Section 2.4 in some synthetic problems. All the numerical results were obtained on a MacBook Pro equipped with an Intel Core i7 processor 3GHz with 8 Gb of RAM running MATLAB (Release 2015a) with its standard settings.

Our tests consisted in solving problems of the form (3.7) derived by applying the reweighted ℓ_1–minimization scheme to the datasets available at `https://github.com/basp-group/co-dmri`; each weighted-$\ell 1$ problem was solved with GP_Ex (Algorithm 7), GP (Algorithm 6), Scaled GP_Ex (Algorithm 8) and SGP (Algorithm 9) methods, considering GP_Ex as the state-of-the-art method to perform comparisons.

A volume of $\Lambda = 16 \times 16 \times 5$ voxels, an unknown FOD of $n = 201$ components, different acquisition signals of dimensions $m_1 = 6$ and $m_2 = 15$ and a maximum estimated number of fibre equal to $\kappa = 3$ in each voxel were considered. This setting led to assess a sequence of problems with dimensions $M_1 = 7680$, $M_2 = 19200$, $N = 257280$ and $K = 771840$. Different noise level corrupting the signals were also considered, in particular we used datasets with SNR = 10, 15, 20.

The experimental setup is summarized as follows. For each problem (3.7)

1. GP_Ex is executed with high accuracy in order to obtain an estimated ground–truth $\boldsymbol{X}^*$, by means of the stopping criterion
$$S(\boldsymbol{X}^k) := \frac{\left\|F(\boldsymbol{X}^{k+1}) - F(\boldsymbol{X}^k)\right\|_2}{\left\|F(\boldsymbol{X}^k)\right\|_2} \leq 7 \cdot 10^{-4} \ ;$$
2. GP_Ex is executed again and stopped with a milder tolerance by means of the criterion
$$S(\boldsymbol{X}^k) \leq 10^{-3}$$
providing a solution $\bar{\boldsymbol{X}}^*$;

3. GP, Scaled GP_Ex and SGP are executed and stopped when

$$F(\boldsymbol{X}^k) \leq F(\bar{\boldsymbol{X}}^*)$$

where $\bar{\boldsymbol{X}}^*$ is the solution issued by GP_Ex at step 2.

In order to evaluate the performance of the methods, we use the following error distances

$$Err(i) := \left\|\Phi\boldsymbol{X}(i) - \boldsymbol{B}\right\|_2 - R^* \ , \quad Err(T) := \left\|\Phi\boldsymbol{X}(T) - \boldsymbol{B}\right\|_2 - R^* \tag{3.18}$$

where $\boldsymbol{X}(i)$ is the approximation of the solution after i iterations, $\boldsymbol{X}(T)$ is the approximation of the solution after T seconds and

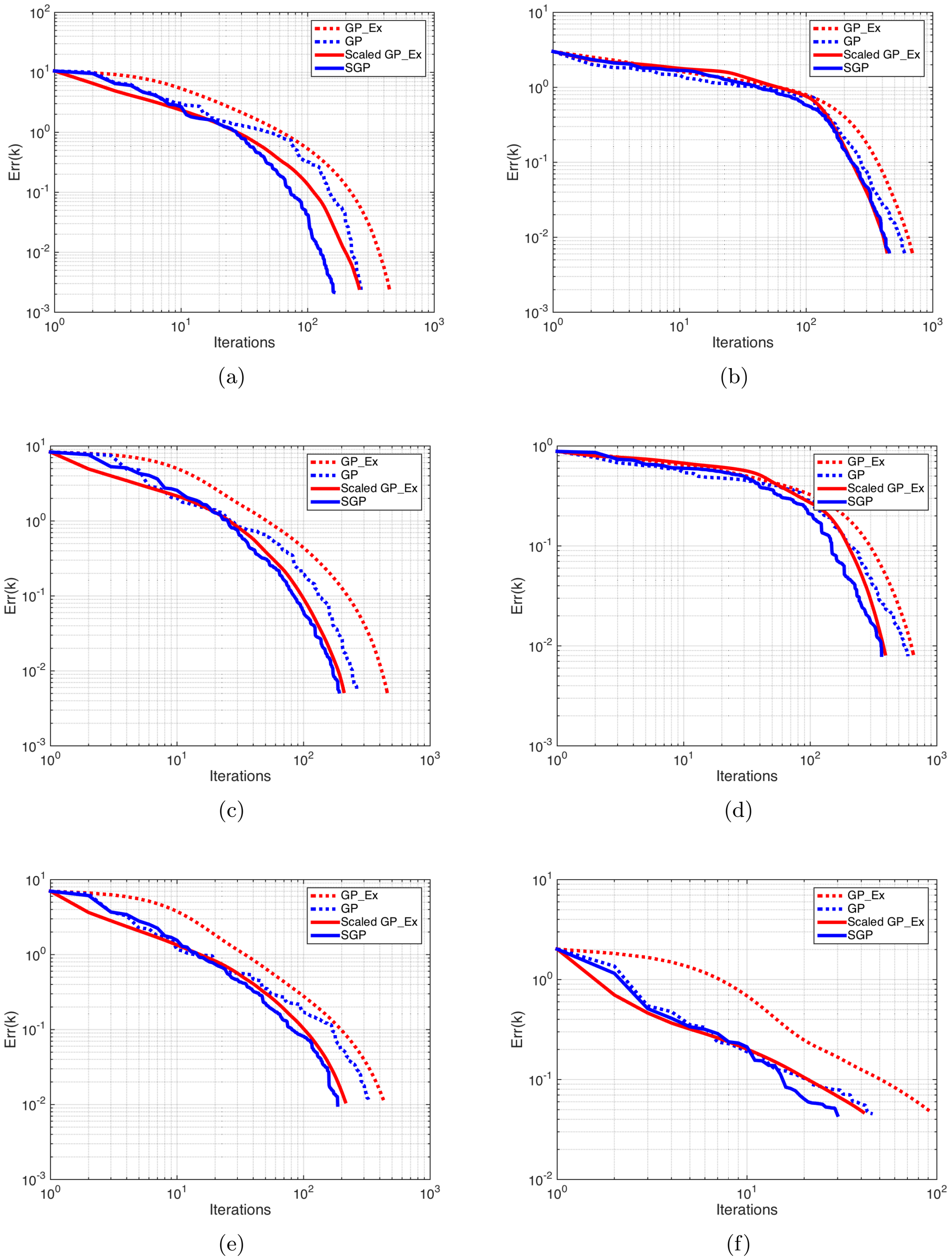
$$R^* = \left\|\Phi\boldsymbol{X}^* - \boldsymbol{B}\right\|_2$$

with $\boldsymbol{X}^*$ corresponding to the solution computed by GP_Ex at step 1. Moreover, the *Residual* quantity shown in Table 3.1 is defined as $\|\Phi\tilde{\boldsymbol{X}} - \boldsymbol{B}\|_2$, where $\tilde{\boldsymbol{X}}$ is the solution computed by the methods.

Figure 3.1 and Figure 3.2 show the behaviour of the considered first–order methods with respect to iterations and time for two different test problems denoted by TP1 and TP2. The results in Table 3.1 and Figures 3.1 and 3.2 show that better performances can be obtained when applying the scaled version of both the considered algorithms. In particular, the time reduction for the fibre orientation estimation problems emphasizes the usefulness of the proposed scaling strategy in solving large–scale dMRI problems.

Test		Algorithm	Residual	Iterations	Comp. $\neq 0$	Time (s)
$m_1 = 6$	SNR = 10 TP1	GP_Ex	8.68775514e+00	439	3141	12.50
		GP	8.68747419e+00	263	2950	9.70
		Scaled GP_Ex	8.68771874e+00	254	2776	5.51
		SGP	8.68735279e+00	160	3214	5.16
$m_1 = 6$	SNR = 10 TP2	GP_Ex	5.68775185e+00	685	5017	23.57
		GP	5.68773235e+00	593	4607	25.95
		Scaled GP_Ex	5.68770834e+00	433	4759	10.51
		SGP	5.68769867e+00	452	5073	15.93
$m_2 = 15$	SNR = 10 TP1	GP_Ex	1.18073857e+01	455	4699	10.87
		GP	1.18073536e+01	265	4557	8.59
		Scaled GP_Ex	1.18073857e+01	207	4384	4.65
		SGP	1.18073048e+01	190	4822	6.15
$m_2 = 15$	SNR = 10 TP2	GP_Ex	1.09324568e+01	649	6591	21.76
		GP	1.09322165e+01	589	6022	24.63
		Scaled GP_Ex	1.09323656e+01	390	6629	9.45
		SGP	1.09320978e+01	365	7014	13.07
$m_2 = 15$	SNR = 15 TP1	GP_Ex	7.83868965e+00	430	6260	10.10
		GP	7.83864069e+00	328	6220	11.99
		Scaled GP_Ex	7.83845331e+00	213	5893	4.88
		SGP	7.83740043e+00	185	6151	6.32
$m_2 = 15$	SNR = 20 TP1	GP_Ex	6.20914135e+00	92	11576	2.27
		GP	6.20761867e+00	44	13574	1.71
		Scaled GP_Ex	6.20887673e+00	40	13168	1.01
		SGP	6.20583557e+00	29	13854	1.03

Table 3.1: FOD tests. From left to right: residual value, number of iterations required to meet the stopping criterion, number of non-negative solution's components and execution time.



(a)

(b)

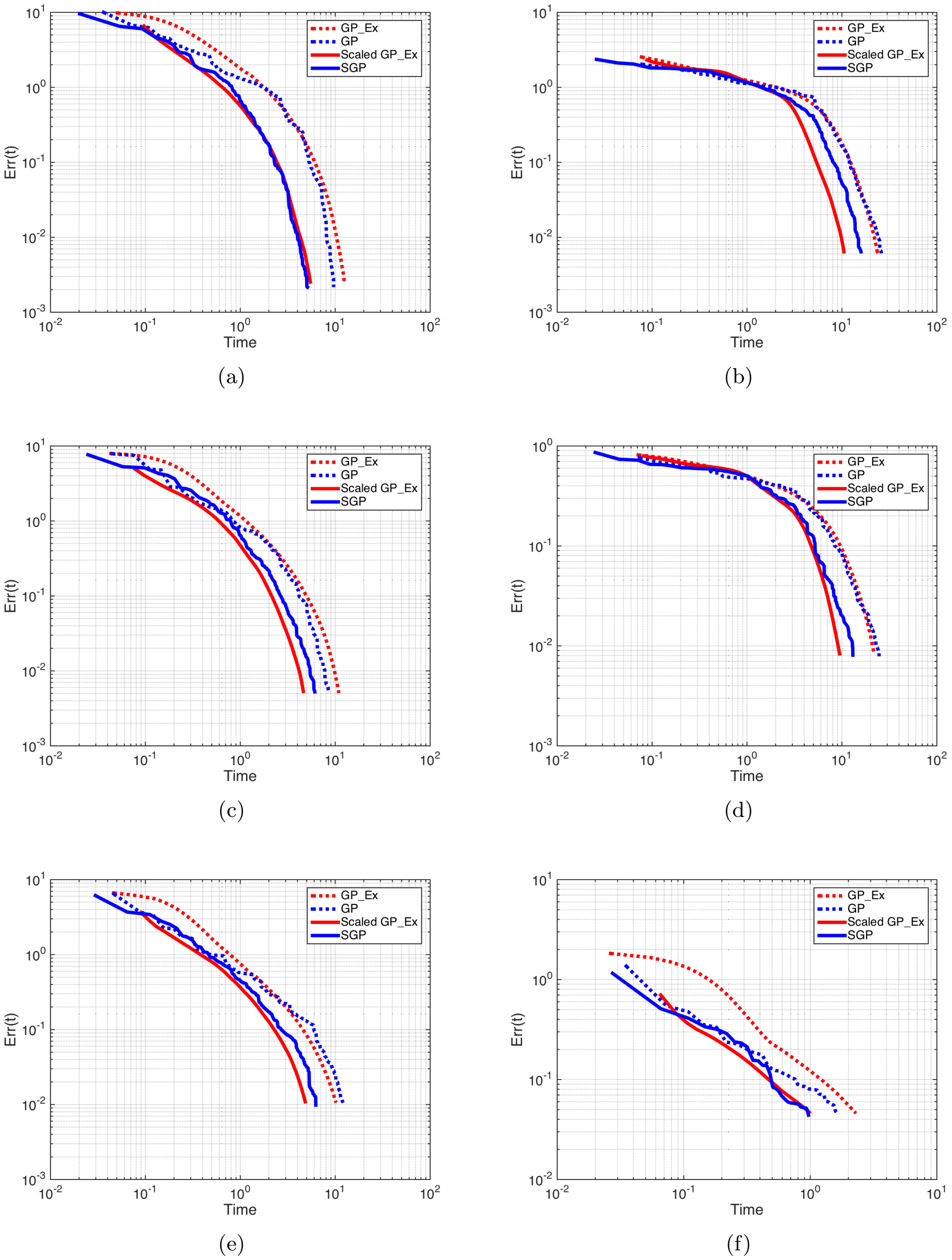
(c)

(d)

(e)

(f)

Figure 3.1: FOD tests, error versus iterations. First row: test problem with $m_1 = 6$ and SNR =10, TP1 (left) and TP2 (right); second row: test problem with $m_2 = 15$ and SNR =10, TP1 (left) and TP2 (right); third row: test problem with $m_2 = 15$, SNR =15, TP1 (left) test problem with $m_2 = 15$, SNR =20, TP1 (right).



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.2: FOD tests, error versus computational time. First row: test problem with $m_1 = 6$ and SNR =10, TP1 (left) and TP2 (right); second row: test problem with $m_2 = 15$ and SNR =10, TP1 (left) and TP2 (right); third row: test problem with $m_2 = 15$, SNR =15, TP1 (left) test problem with $m_2 = 15$, SNR =20, TP1 (right).

Chapter 4

Application to 3D Computed Tomography

In this chapter we consider an optimization problem arising in 3D X–ray Computed Tomography (CT) image reconstruction [22, 84] from low sampling acquisitions, i.e., when the CT system acquires only a reduced set of data. This application has recently received growing attention in the medical community, since sub–sampling acquisitions have several advantages over the traditional complete sampling acquisitions in speeding up the imaging process, thus reducing the exposure to ionizing radiations and increasing the patient safety [21, 61, 94].

In these cases the traditional analytical reconstruction methods such as the Filtered Back–Projection (called Feldkamp method in 3D [51]) produce images of low quality, with extreme artifacts and high noise. Iterative Image Reconstruction (IIR) methods are generally preferred in this framework because they can introduce a priori information about the unknown object and they can exploit the Compressive Sampling (CS) theory [24] for reconstructing a signal or an image from a reduced number of acquisitions with respect to the Nyquist theory [61]. The drawback of IIR algorithms is their higher computational cost with respect to the analytical methods, but thanks to the dramatic improvement of CPUs speed and the possibility to perform parallel computation at low cost on GPUs, the time for the IIR algorithms execution is now acceptable even in the clinical setting [8].

In real applications, the problem has a very large size, of the order of billions, and the problem solution is very challenging, because in the clinical applications a good image must be reconstructed in at most 1–2 minutes. For these reasons, the IIR methods are not executed until convergence, but they are stopped after the desired time. Thus, it is essential to use a reconstruction algorithm with non–expensive iterations and fast convergence in the very first iterations. To this end, some suggestions are available in the recent literature: first–order optimization methods based on accelerated gradient schemes have been proposed in [69, 105, 108], optimization transfer methods have been applied to CT image reconstruction for example in [46, 72, 74], a momentum approach based on Nesterov method [88] can be found in [73] and a Fixed Point method using approximate second–order information has been used in [78].

The aim of our analysis is to investigate the efficiency of an SGP approach in the considered CT application. We evaluate the behaviour of the proposed SGP algorithm in comparison with a standard non–scaled GP method and the state–of–the–art accelerated gradient method [69], which has been successfully applied in tomographic reconstruction problems.

The chapter is organized as follows. In Section 4.1 we describe the 3D CT discrete model and we formulate the constrained optimization problem; in Section 4.2 we present the different settings implemented for the SGP algorithm and in Section 4.3 we show the numerical results obtained on a 3D phantom.

4.1 Problem formulation

In a 3D cone beam tomography system a cone of X–rays is emitted by a source rotating along an arc or a circular trajectory around the object of interest (see Figure 4.1) from a fixed number of positions (or angles). The rays, partially absorbed by the object, are projected on a flat panel detector (that can possibly move with the source) and then recorded.

Following the Lambert–Beer's model that relates the recorded value $b_i^{(\theta)}$ at each pixel i of the detector, for a fixed angle θ, with the attenuation coefficient μ at each position $\mathbf{w}$ of the object we obtain the image formation model for X–rays tomographic images; in details,

$$b_i^{(\theta)} = \mathtt{exp}\left(-\int_{L_\theta} \mu(\mathbf{w})d\ell\right) + \eta_i^{(\theta)}, \qquad i = 1, \ldots, N_p, \quad \theta = 1, \ldots, N_\theta, \tag{4.1}$$

where N_p is the number of pixels in the detector, N_θ is the number of angles, L_θ is the line followed by the X–ray beam through the object, $\mu(\mathbf{w})$ is the linear attenuation coefficient at the position $\mathbf{w}$, depending on the material in the object and characterizing the structures inside the object, and, lastly, $\eta_i^{(\theta)}$ is the noise measured at the detector (pixel i, angle θ) and it includes scattering and electronic noise.

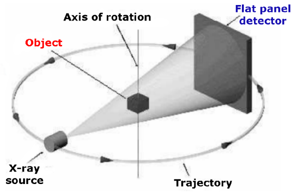


Figure 4.1: scheme of a 3D X–rays CT tomography scanning geometry.

The IIR methods consider the discrete linearization of (4.1):

$$Ax = b \tag{4.2}$$

where $b \in \mathbb{R}^{N_p \times N_\theta}$ $(b > 0)$ is the vector of recorded projections affected by noise, $x \in \mathbb{R}^{N_v}$ represents the discretization of $\mu(\mathbf{w})$ in the N_v voxels of the object (lexicographically ordered in a vector) and $A \in \mathbb{R}^{(N_p \times N_\theta) \times N_v}$ is the matrix describing the system geometry. In real applications, N_p is of the order of millions and N_v is of order of few billions and N_θ is of order 10^1 for sparse tomography. Different algorithms can be found in literature for the computation of A; we use here the Siddon algorithm [104] based on geometrical ray–tracing. Each element $a_{i,j}^{\theta}$ of A represents the length of the intersection of the ray, emitted at angle θ, recorded by pixel i of the detector, with the voxel j (in this notation, the pixels of the detector and the voxels of the object are lexicographically ordered in vectors). Moreover, in a reasonable physical setting, the matrix A has elements greater than zero in each column, because each voxel is projected at least once onto the detector. In the case of reduced sampling acquisition here considered, $N_p \times N_\theta < N_v$ hence the linear system (4.2) has infinite possible solutions. Furthermore, since the linear system comes from the discretization of the integral equation of the first kind (4.1), the problem is ill–conditioned and some of the solutions of (4.2) are dominated by noise; thus, regularization strategies are necessary.

The problem can be reformulated as a penalized optimization problem of the form [106]:

$$\min_{x \geq 0} f(x) = J(x) + \lambda R(x) \tag{4.3}$$

where $J(x)$ is the fit–to–data function involving the discretization of the continuous integral imaging model whose expression is related to the kind of noise on the data; $R(x)$ is a regularization function and λ is the regularization parameter.

Since tomographic data are affected by mixed Gaussian and Poisson noise, it is desirable to be able to efficiently solve problem (4.3) with $J(x)$ equal to the linear Least Squares (LS) functional, when Gaussian noise is dominant, or equal to the nonlinear Kullback–Leibler (KL) divergence, when Poisson noise is dominant. The possibility to choose the proper form of $J(x)$ lies on the real application necessities, as the dominant kind of noise depends on the specific physical CT system. The function $R(x)$ should reduce the noise, regularize the ill–conditioned problem arising from the discretization of an ill–posed Fredholm integral equation and impose some sparsity on the computed solution following the CS theory. Since many medical images are almost uniform inside the organs, the most widely used regularization function for the CT problems is the Total Variation (TV) function [64, 107, 109, 106, 105, 108, 116, 73, 101]

$$TV(x) = \int_{\Omega} |\nabla x| dx \tag{4.4}$$

that forces the sparsity in the gradient domain of the solution.

Then, the problem can be reformulated as a penalized optimization problem of the form [106]:

$$\min_{x \geq 0} f(x) = J(x) + \lambda TV_\beta(x) \tag{4.5}$$

where $TV_\beta(x)$ is a smoothed differentiable version of the TV function defined as [115]:

$$TV_\beta(x) = \sum_{j=1}^{N_v} \sqrt{\|\nabla x_j\|_2^2 + \beta^2} \tag{4.6}$$

with β a positive small parameter.

For what concerns the fit–to–data function $J(x)$, its expression is related to the noise on the data. Following a Maximum Likelihood approach [11], if the noise has a Gaussian distribution, the Least Squares function:

$$J(x) = \frac{1}{2}\|Ax - b\|_2^2 \tag{4.7}$$

gives the appropriate fit–to–data function, while, if the noise has a Poisson distribution, the Kullback–Leibler divergence

$$J(x) = \sum_{i=1}^{N_p \times N_\theta} \left(\sum_{j=1}^{N_v} A_{ij}x_j + bg - b_i - b_i log \frac{\sum_{j=1}^{N_v} A_{ij}x_j + bg}{b_i} \right) \tag{4.8}$$

($bg > 0$ is the background value) is the suitable term. The noise on the CT data is mixed Poisson (due to the X–rays particles behaviour) and Gaussian (due to the recording digital system) and the dominant one depends on the particular system considered. Hence, we consider in this paper the two different cases in which the fit–to–data term $J(x)$ is defined as in (4.7) or as is (4.8). In both cases, the objective function of the problem (4.5) is coercive and strictly convex on the nonnegative orthant, therefore the problem has a unique solution.

Problem discretization

For the discussion in the next section, it is useful to introduce the discretization of the problem in the 3D setting, by using the notation j_x, j_y, j_z to indicate the indices of a voxel of the discrete object on the three cartesian axes.

The $TV_\beta(x)$ function is discretized by forward differences with boundary periodic conditions. Starting from works [115, 119] which deal with bi–dimensional problems, we derive the expression of the discrete $TV_\beta(x)$ function in the 3–dimensional case [77]:

$$TV_\beta(x) := \frac{1}{2} \sum_{j_x=1}^{N_x} \sum_{j_y=1}^{N_y} \sum_{j_z=1}^{N_z} \phi(\delta^2 x_{j_x,j_y,j_z}) \tag{4.9}$$

where $N_x \times N_y \times N_z = N_v$,

$$\delta^2 x_{j_x,j_y,j_z} = (x_{j_x+1,j_y,j_z} - x_{j_x,j_y,j_z})^2 + (x_{j_x,j_y+1,j_z} - x_{j_x,j_y,j_z})^2 + (x_{j_x,j_y,j_z+1} - x_{j_x,j_y,j_z})^2$$

and

$$\phi(t) = 2\sqrt{t + \beta^2}.$$

In order to better explain some details of the SGP algorithm presented in the next section, it is convenient to recall also the form of the (j_x, j_y, j_z) entry of the gradient of $TV_\beta(x)$:

$$\begin{aligned}
\frac{\partial TV_\beta}{\partial x_{j_x,j_y,j_z}}(x) &= \frac{1}{2}\frac{\partial}{\partial x_{j_x,j_y,j_z}}\left(\phi(\delta^2 x_{j_x,j_y,j_z}) + \phi(\delta^2 x_{j_x-1,j_y,j_z}) + \phi(\delta^2 x_{j_x,j_y-1,j_z}) + \phi(\delta^2 x_{j_x,j_y,j_z-1})\right)\\
&= \left(\phi'(\delta^2 x_{j_x,j_y,j_z})(3x_{j_x,j_y,j_z} - x_{j_x+1,j_y,j_z} - x_{j_x,j_y+1,j_z} - x_{j_x,j_y,j_z+1})\right) +\\
&+ \left(\phi'(\delta^2 x_{j_x-1,j_y,j_z})(x_{j_x,j_y,j_z} - x_{j_x-1,j_y,j_z})\right) +\\
&+ \left(\phi'(\delta^2 x_{j_x,j_y-1,j_z})(x_{j_x,j_y,j_z} - x_{j_x,j_y-1,j_z})\right) +\\
&+ \left(\phi'(\delta^2 x_{j_x,j_y,j_z-1})(x_{j_x,j_y,j_z} - x_{j_x,j_y,j_z-1})\right).
\end{aligned}$$

4.2 Optimization methods

In this section we describe the updating rules for the scaling matrix D_k and the steplength α_k that allow SGP (Section 2.4.2, Algorithm 9) to efficiently solve the described CT problem. For sake of clarity, we report here the general iteration of the SGP method for the solution of problem (4.5)

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \lambda_k \left(P_+(\boldsymbol{x}^{(k)} - \alpha_k D_k \nabla f(\boldsymbol{x}^{(k)})) - \boldsymbol{x}^{(k)}\right). \tag{4.10}$$

recalling the fact that a classic GP scheme can be obtained by choosing $D_k = I_n$. As the feasible set of problem (4.5) is $\Omega = \left\{\boldsymbol{x} \in \mathbb{R}^{N_v} | \boldsymbol{x} \geq \boldsymbol{0}\right\}$, $P_+(z)$ is the euclidean projection of the vector $z \in \mathbb{R}^{N_v}$ onto the nonnegative orthant.

Following the technique described in Section 2.4.2, we define the diagonal scaling matrix by means of special splittings of the gradient of the objective function:

$$\nabla f(x) = V^f(x) - U^f(x), \qquad V^f(x) > 0, \quad U^f(x) \geq 0, \tag{4.11}$$

where $V^f(x)$ and $U^f(x)$ are obtained as:

$$V^f(x) = V^J(x) + \lambda V^{TV}(x), \quad U^f(x) = U^J(x) + \lambda U^{TV}(x),$$

with

$$\begin{aligned}
\nabla J(x) &= V^J(x) - U^J(x), & V^J(x) > 0, \quad U^J(x) \geq 0,\\
\nabla TV_\beta(x) &= V^{TV}(x) - U^{TV}(x), & V^{TV}(x) > 0, \quad U^{TV}(x) \geq 0.
\end{aligned}$$

Given the splitting (4.11), we propose to update the diagonal scaling matrix $D_{k+1} = \{d_{j,j}^{(k+1)}\}$ in the following way:

$$d_{j,j}^{(k+1)} = \min\left(\rho_{k+1}, \max\left(\frac{1}{\rho_{k+1}}, \frac{x_j^{(k+1)}}{V_j^f(x^{(k+1)})}\right)\right), \qquad j = 1, \ldots, n.$$

The vectors $V^J(x)$ and $V^{TV}(x)$ defining $V^f(x)$ are set by taking into account the special form of $\nabla J(x)$ and $\nabla TV_\beta(x)$, respectively.

When the $J(x)$ is the LS function (4.7), the gradient of the fit–to–data term has the form:

$$\nabla J(x) = A^T Ax - A^T b;$$

in this first case, we choose:

$$V^J(x) = A^T Ax, \quad U^J(x) = A^T b.$$

When the fit–to–data term $J(x)$ is the KL function (4.8), we have that:

$$\nabla J(x) = A^T \mathbf{1} - A^T Y^{-1} \boldsymbol{b},$$

where $\mathbf{1} \in \mathbb{R}^n$ is a vector whose components are all equal to one and $Y = \text{diag}\ (A\boldsymbol{x} + bg)$ is a diagonal matrix with the entries of $(A\boldsymbol{x} + bg)$ on the main diagonal; in this second case we choose:

$$V^J(x) = A^T \mathbf{1}, \quad U^J(x) = A^T Y^{-1} b.$$

Accordingly with the notation introduced in Section 4.1 for the entries of $\nabla TV_\beta(x)$, we finally set the components of $V^{TV}(x)$ as

$$\begin{aligned} V^{TV}_{j_x,j_y,j_z}(x) &= \big(3\phi'(\delta^2 x_{j_x,j_y,j_z}) + \phi'(\delta^2 x_{j_x-1,j_y,j_z}) + \\ &\quad \phi'(\delta^2 x_{j_x,j_y-1,j_z}) + \phi'(\delta^2 x_{j_x,j_y,j_z-1})\big)\, x_{j_x,j_y,j_z}. \end{aligned}$$

Following the suggestions in [17, 31, 30], the parameter ρ_{k+1} is chosen as $\rho_{k+1} = \sqrt{1 + 10^{15}/(k+1)^{2.1}}$.

Steplength selection

Once the scaling matrix D_{k+1} has been defined, a new value for the steplength α_{k+1} can be computed with the aim to achieve further acceleration of the iterative process. We equipped the SGP algorithm with two steplength selection strategies. First, the alternating strategy similar to (2.41) (introduced in Section 2.3.2) with variable parameter τ_k is used:

$$\alpha_{k+1} = \begin{cases} \min\left\{\alpha_j^{BB2} :\ j = \max\{1, k+1-m_\alpha\}, \ldots, k+1\right\}, & \text{if } \frac{\alpha_{k+1}^{BB2}}{\alpha_{k+1}^{BB1}} < \tau_k \\ \alpha_{k+1}^{BB1}, & \text{otherwise} \end{cases} \tag{4.12}$$

where $m_\alpha = 2$, $\tau_0 = 0.5$ and the parameter τ_k is updated in the following way

```
if $\alpha_{k+1}^{\text{BB2}}/\alpha_{k+1}^{\text{BB1}} < \tau_k$
    $\tau_{k+1} = 0.9\,\tau_k$,
else
    $\tau_{k+1} = 1.1\,\tau_k$,
end.
```

Secondly, the so called Ritz–like values in described in Section 2.4.2 [Steplength choice], are used to define the steplengths for $m = 3$ iterations:

$$\alpha_{k+j} = \frac{1}{\theta_j},\ j = 1, 2, 3. \tag{4.13}$$

4.3 Numerical experiments

In this section we present the numerical results performed on on a a MacBook Pro equipped with an Intel Core i7 processor 3GHz with 8 Gb of RAM running MATLAB (Release 2015a) with its standard settings. For performing the tests, we used some functions of the `TVReg` Matlab Toolbox, `http://www.imm.dtu.dk/~pcha/TVReg/` [71].

4.3.1 Test problem

We consider as the true object x^* the digital Shepp Logan phantom discretized in $N_v = N_x \times N_y \times N_z = 61 \times 61 \times 61 = 226981$ voxels lexicographically ordered in a vector. The slices number 24, 31 and 35 in the z direction are shown in Figure 4.2. The projections have been created as:

$$b^* = A \cdot x^*$$

where A is the projection matrix, obtained with the functions in the `TVReg` Toolbox which simulates a system where a source moves on a semi–sphere emitting X–rays cone beams from N_θ angles. The detector is supposed with $N_p = 61 \times 61$ pixels and the number of angles N_θ varies in the set $\{19, 37, 55\}$. In all the cases the problem is underdetermined. The projections are corrupted by noise, with both Gaussian and Poisson distribution, as specified in the following subsections.

Stopping criterion and parameters

We describe here the stopping criterion for the SGP algorithm and the setting of its main parameters. If k is the index of the current iteration and

$$S_k^f := \frac{|f(x^{(k+1)}) - f(x^{(k)})|}{|f(x^{(k)})|}$$

is the relative distance between successive values of the objective function, we consider the conditions

$$S_k^f \le \epsilon_1, \qquad \frac{1}{p}\sum_{j=0}^{p-1} S_{k-j}^f \le \epsilon_2 \ \text{ if } \ k \ge p-1,$$

where $\epsilon_1 = 10^{-6}$, $p = 20$ and $\epsilon_2 = 10^{-5}$; the SGP stopping criterion consists in satisfying both the conditions or performing a maximum number of $k = 1000$ iterations.

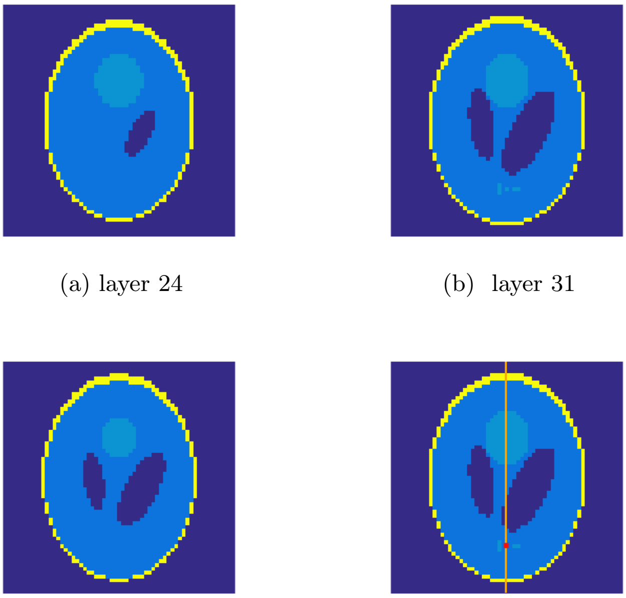
For what concerns the SGP parameters, the setting reported below is used:

- $\gamma = 0.4$ and $\sigma = 10^{-4}$ as backtracking parameters;
- $\alpha_{min} = 10^{-10}$, $\alpha_{max} = 10^5$, $\alpha_0 = 1$, $m_\alpha = 2$ and $\tau_0 = 0.5$ for the steplength selection.

Results evaluation

In order to evaluate the reconstruction results, we consider the the Relative Error (*Relerr*) between the exact volume x^* and the reconstructed image $\tilde{x}$ ($Relerr = \|x^* - \tilde{x}\|_2 / \|x^*\|_2$). The reconstructed images are also evaluated by plotting the profile of the yellow vertical line in Figure 4.2d, vertical profile (VP), and the profile over the 61 layers in the z direction of the red pixel in Figure 4.2d, depth profile (DP).

We show the results obtained by the algorithms at three different temporal windows: at 5 seconds (10–15 iterations), for simulating a real–time execution; at 20 seconds (50–60 iterations), corresponding to an over–time of few minutes in real applications; at the convergence, i.e., when the convergence criterium is satisfied (this is a long execution that can be performed only off–line in a real application). Each of these three different outputs reflects a practical interest and together they represent the evolution of the methods in time.


(a) layer 24

(b) layer 31

(c) layer 35

(d) layer 31

Figure 4.2: different layers in the z–direction of the original phantom. For the analysis of the results, in (d) some interesting features in layer 31 are highlighted: a yellow line along which we analyse the vertical profile and a red pixel to examine the depth profile.

4.3.2 Results for Gaussian noise tests

In this paragraph we show the results obtained on the simulated data $b = b^* + e$, where e is the vector representing white Gaussian noise with level defined as $\nu = \frac{\|e\|_2}{\|b^*\|_2}$; we consider here $\nu = 0.01$, corresponding to a Signal–to–Noise Ratio SNR $:= 20 \cdot log_{10}(\frac{\|b\|_2}{\|b-b^*\|_2})$ of about 40. We consider the fit–to–data function $J(x)$ as the LS function and we set the TV smoothing parameter β equal to 0.001 in all the experiments. The regularization parameter λ has been heuristically set to 0.09; we have experimented that for this test the model is not very sensitive to the value of λ (similar results have been obtained with different values of λ in the interval $[0.005, 0.5]$).

We compare the results obtained by the proposed SGP method equipped with ABB steplengths (4.12), the non–scaled GP method (GP) equipped with the same steplength selection used by SGP and the Unknown Parameter Nesterov (UPN) method proposed in [69], implemented in the `TVReg` toolbox. The UPN method has been equipped with the same stopping criterion used for GP and SGP and its parameters have been set at their best values after a careful tuning.

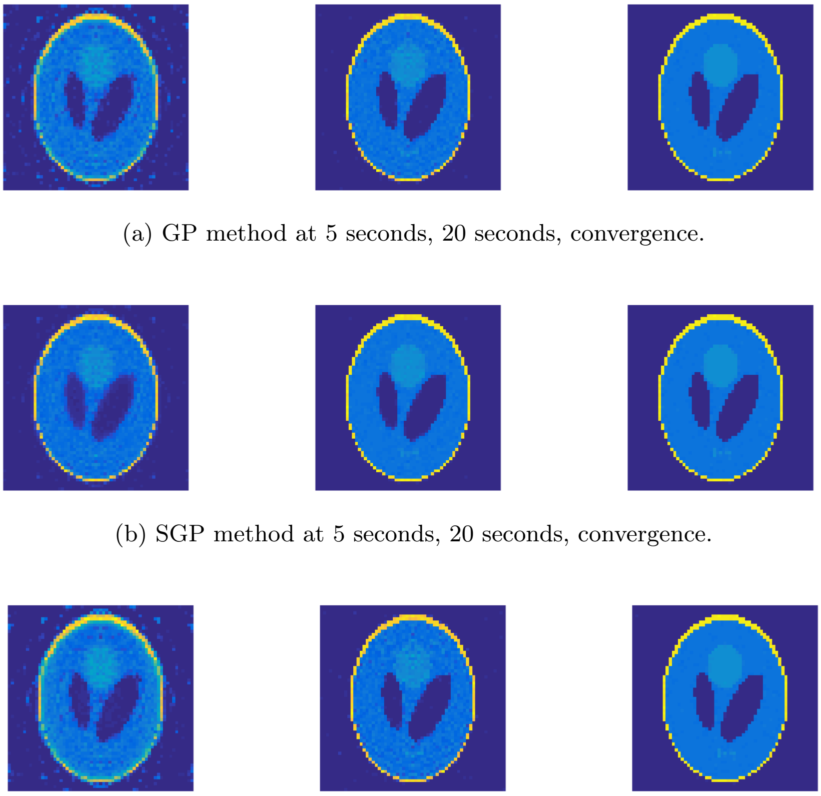
In Table 4.1 we present the results obtained with different number of views ($N_\theta = 19, 37, 55$) for the GP, SGP and UPN methods. In the columns from left to right we report the Relative Error, the objective function value and the number of performed iterations in the three considered temporal windows: at 5 seconds, at 20 seconds and at convergence. From the table, we see that the SGP method outperforms the others in the first iterations (5 and 20 seconds) for all the considered angles; at convergence, all the methods give very similar results. The reconstructions of central layer (layer 31) obtained with the three considered methods in the case $N_\theta = 37$ are shown in Figure 4.3.

In Figure 4.4 the errors versus the iterations (on the left) and the objective function values versus the iterations (on the right) are shown in log–log scale. We compare here the GP method (blue line), the SGP method (red line) and the UPN method (green line) up to the convergence of the methods. The advantage of using the scaling matrix is evident, especially in the first iterations.

Figure 4.5 displays the VP (on the left) and DP (on the right) after 5 seconds, 20 seconds and at convergence. We compare again the GP reconstruction (blue line), SGP reconstruction (red line) and UPN reconstruction (green line) with the phantom profile (grey line). The VP plots confirm that after few iterations (5 seconds) we can identify, in the signal reconstructed by the SGP method, all the objects with a good approximation of their intensity; in the DP plot after 20 seconds the SGP method has almost completely eliminated the noise, while the GP and UPN plots show a residual noise yet. We can see that the SGP profiles are less noisy than the others and in the DP the peak of the SGP line is the closest to the exact one.

			Relerr	fun	iters
$N_\theta = 19$	GP	5 s	0.3816	5574.36	16
		20 s	0.1548	1596.75	69
		conv	0.0559	1498.49	263
	SGP	5 s	0.2637	3396.96	19
		20 s	0.1178	1560.48	71
		conv	0.0543	1498.57	198
	UPN	5 s	0.3785	6075.55	11
		20 s	0.1786	1652.62	48
		conv	0.0580	1484.46	606
$N_\theta = 37$	GP	5 s	0.3475	11537.90	16
		20 s	0.0898	1795.02	64
		conv	0.0245	1645.77	154
	SGP	5 s	0.1840	4335.56	18
		20 s	0.0477	1689.30	66
		conv	0.0247	1646.39	194
	UPN	5 s	0.4001	19918.90	8
		20 s	0.1045	1917.49	46
		conv	0.0241	1632.07	224
$N_\theta = 55$	GP	5 s	0.3091	14306.80	15
		20 s	0.0779	1997.80	60
		conv	0.0199	1783.11	142
	SGP	5 s	0.2148	9662.70	16
		20 s	0.0277	1814.38	60
		conv	0.0199	1783.60	147
	UPN	5 s	0.4315	40865.00	6
		20 s	0.0677	2033.66	46
		conv	0.0199	1769.47	200

Table 4.1: results obtained on the test problems with data affected by Gaussian noise.



(a) GP method at 5 seconds, 20 seconds, convergence.

(b) SGP method at 5 seconds, 20 seconds, convergence.

(c) UPN method at 5 seconds, 20 seconds, convergence.

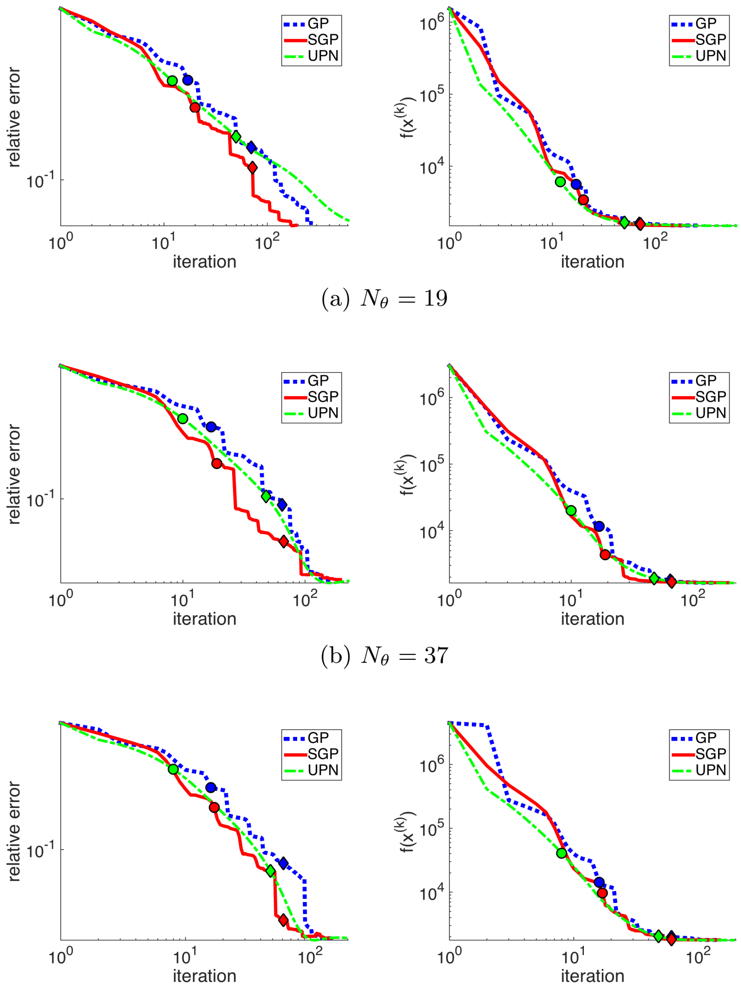
Figure 4.3: reconstructions obtained in case of Gaussian noise on the data. From the left to the right: reconstructions after 5 seconds, after 20 seconds, at convergence.

4.3.3 Results for Poisson noise tests

We consider now some tests where the projections are affected by Poisson noise, with SNR $\simeq 40$ and background $bg = 10^{-5}$. The problem is solved by using the KL fit–to–data function in (4.8). In this case the regularization parameter λ has been heuristically set to 0.03 and the TV smoothing parameter $\beta = 0.01$; we have experimented that, as in the case of Gaussian noise, similar results have been obtained with different values of λ in the interval $[0.001, 0.1]$.

In order to test the effectiveness of the acceleration strategies proposed in Sections 2.3 and 2.4, we compare the results obtained with the GP and the SGP methods on four distinct implementations, based on either the ABB rules or the Ritz–like values: in the rest of this Chapter, GP ABB and SGP ABB will refer to GP (Algorithm 6) and SGP (Algorithm 9) methods equipped with ABB steplengths (4.12), while GP R and SGP R will refer to GP and SGP methods equipped with Ritz–like steplengths (4.13).

Table 4.2 reports the results in the case $N_\theta = 19, 37, 55$, with the same information of Table



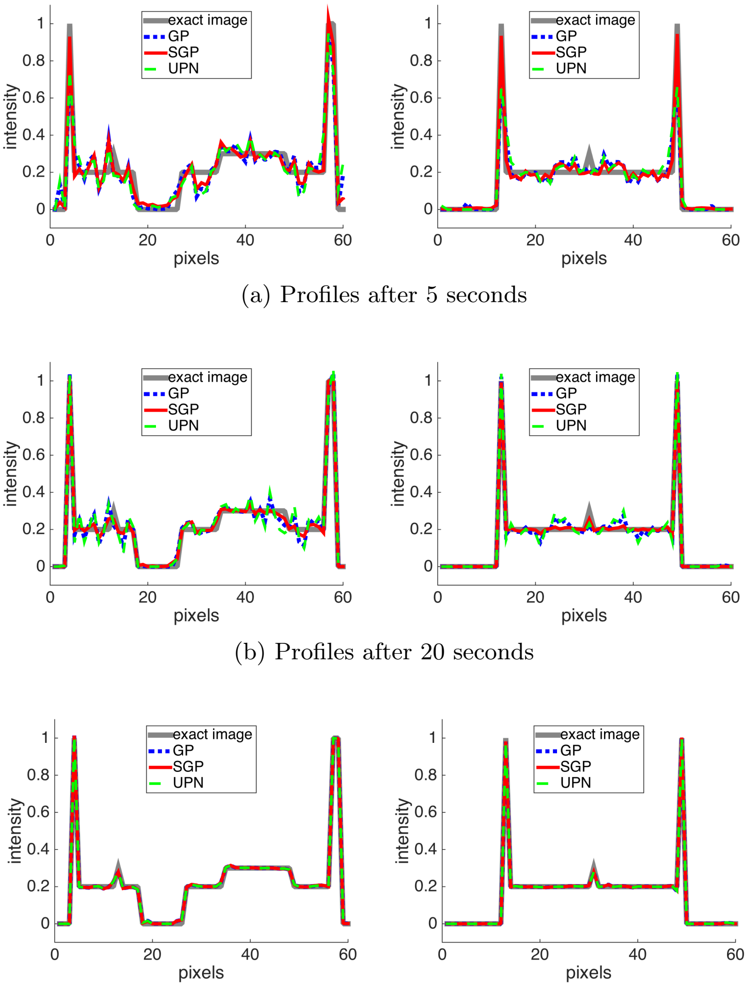
(a) $N_\theta = 19$

(b) $N_\theta = 37$

(c) $N_\theta = 55$

Figure 4.4: case of Gaussian noise. On the left: errors vs iterations; on the right: function values vs iterations. The circles and the diamonds represent the values at 5 and 20 seconds, respectively.

4.1. For the KL model, the performance improvement due to the scaling is more consistent than in the LS model, as it can be seen by the Relative Error values. If the number of performed iterations in the last column is equal to 1000 an asterisk reminds that the algorithm has stopped after reaching the maximum number of iterations. We want to stress that this happens only for the GP method, confirming its slower convergence rate. Figure 4.7 shows the Relative Error versus the iterations in the left panel for GP ABB (black line), GP R (green line), SGP ABB (blue line) and SGP R (red line); the objective function values versus the iterations are displayed in the right panel with the same color correspondence. Independently of the steplength rule, the



(a) Profiles after 5 seconds

(b) Profiles after 20 seconds

(c) Profiles at convergence

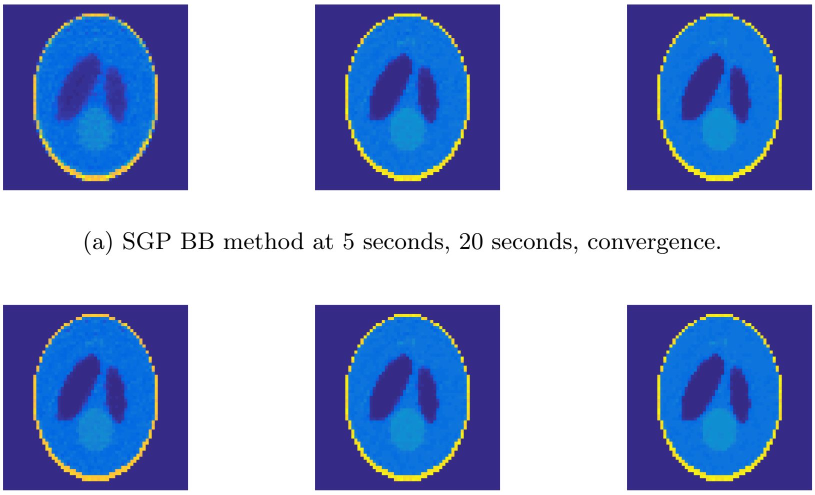
Figure 4.5: case of Gaussian noise. Profiles for 37 angles: on the left VP plots and on the right DP plots at different temporal windows.

scaling strategy accelerates the GP methods considerably, especially in the first iterations. In Figure 4.6 the reconstructions of the layer 31 obtained with SGP equipped with both steplength selection rules after 5 seconds, 20 seconds and at convergence are represented: the quality of the SGP R reconstruction after 5 iterations is noticeable.

The analysis of VP and DP profiles for SGP ABB and SGP R in the case of $N_\theta = 37$ in Figure 4.8 shows that the scaling allows recovering very good profiles in very short time: after 20 seconds the line of the reconstructed image almost overlap the line of the exact phantom (the only exception is the small peak in the center of the DP).

			Relerr	fun	iters
$N_\theta = 19$	GP BB	5 s	0.6301	16683.45	12
		20 s	0.5691	9050.69	61
		conv	0.4480	3402.91	1000*
	GP R	5 s	0.4938	7287.88	14
		20 s	0.3599	4116.67	65
		conv	0.3107	1660.81	882
	SGP BB	5 s	0.2145	768.06	19
		20 s	0.1003	524.97	72
		conv	0.0917	522.81	184
	SGP R	5 s	0.1545	563.89	19
		20 s	0.0976	524.51	71
		conv	0.0870	521.99	206
$N_\theta = 37$	GP BB	5 s	0.6914	55825.21	12
		20 s	0.6396	32657.93	51
		conv	0.4603	7504.14	560
	GP R	5 s	0.4386	12952.16	13
		20 s	0.4250	7810.99	62
		conv	0.3876	4348.95	375
	SGP BB	5 s	0.1831	1201.56	17
		20 s	0.0663	559.66	65
		conv	0.0482	554.39	332
	SGP R	5 s	0.1067	625.84	19
		20 s	0.0580	559.66	68
		conv	0.04851	552.12	252
$N_\theta = 55$	GP BB	5 s	0.6919	85029.32	12
		20 s	0.5186	19534.21	53
		conv	0.4444	9959.44	498
	GP R	5 s	0.6329	73328.95	9
		20 s	0.6305	70221.83	57
		conv	0.293	2992.25	523
	SGP BB	5 s	0.1745	1572.84	16
		20 s	0.0862	703.82	59
		conv	0.0495	595.30	617
	SGP R	5 s	0.1089	727.70	16
		20 s	0.0465	586.32	57
		conv	0.0365	575.86	241

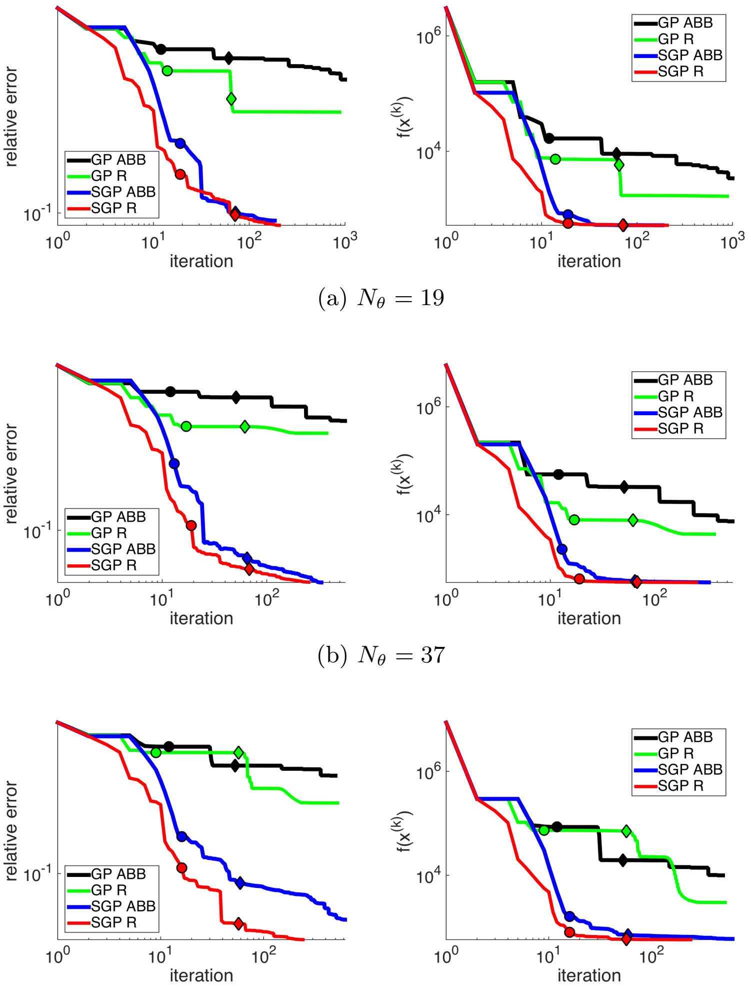
Table 4.2: results obtained on the test problems with data affected by Poisson noise.



(a) SGP BB method at 5 seconds, 20 seconds, convergence.

(b) SGP R method at 5 seconds, 20 seconds, convergence.

Figure 4.6: reconstructions obtained in case of Poisson noise on the data. From the left to the right: reconstructions after 5 seconds, after 20 seconds, at convergence.

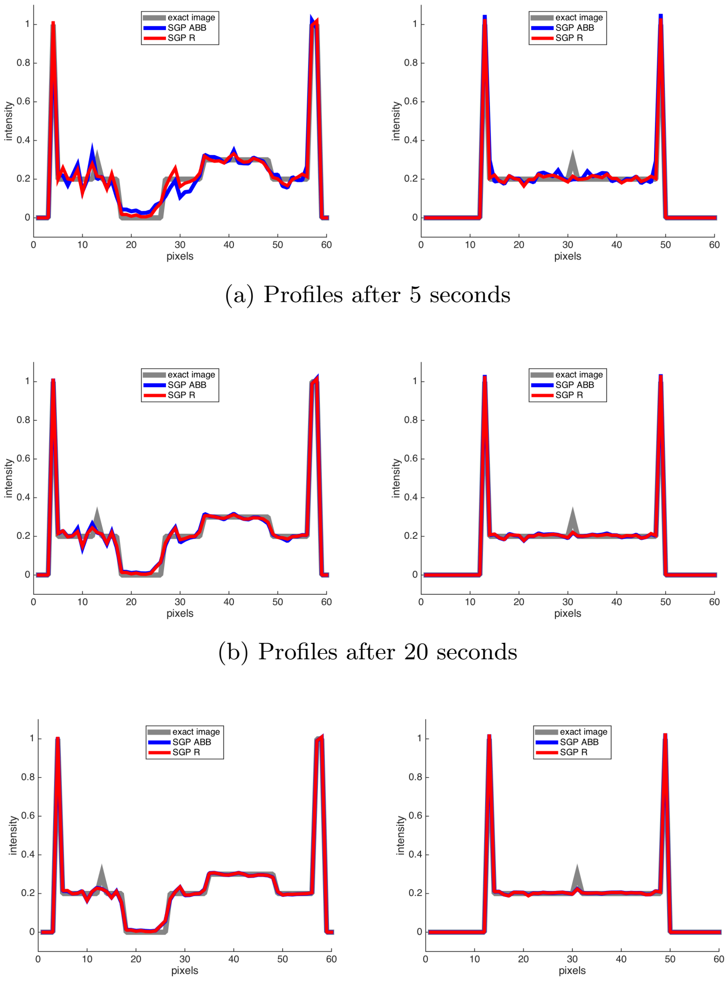


(a) $N_\theta = 19$

(b) $N_\theta = 37$

(c) $N_\theta = 55$

Figure 4.7: case of Poisson noise. On the left: errors vs iterations; on the right: function values vs iterations. The circles and the diamonds represent the values at 5 and 20 seconds, respectively.



(a) Profiles after 5 seconds

(b) Profiles after 20 seconds

(c) Profiles at convergence

Figure 4.8: case of Poisson noise. Profiles for 37 angles: on the left VP plots and on the right DP plots at different temporal windows.

Conclusions

The research activity presented in this thesis has dealt with the analysis of acceleration techniques for first–order methods in nonlinear constrained optimization and their impact in signal reconstruction problems arising in the biomedical domain. The work mainly concerned the design of suitable variable metric strategies induced by scaling matrices and choices of the steplength parmeter for a classical gradient projection method and for a gradient projection method with extrapolation step.

Further work was devoted to the study of regularization methods for inverse problems, with the aim of extending the aforementioned acceleration techniques to a wider class of problems.

The presented methods have been tested on two biomedical imaging problems.

The first application concerned the reconstruction of fibre orientation distribution on the cerebral white matter from diffusion Magnetic Resonance Imaging data, designed as a constrained Least Squares problem with nonnegativity and sparsity constraints. The methods Scaled Gradient Projection equipped with Adaptive Barzilai–Borwein steplength selection rule and the Scaled Gradient Projection with Extrapolation were engaged to find an optimal solution for the problem and they showed competitive performances with respect to the state–of–the–art FISTA algorithm.

The second experimental framework dealt with an image reconstruction problem of 3D X-ray tomography from limited data. In this case, the problem is formulated as the nonnegatively constrained minimization of an objective function expressed by the sum of a fit–to–data term and a smoothed Total Variation function. The choice of the fit–to–data function is strictly related to the noise that affects real Computed Tomography systems; thus different functionals were considered in order to evaluate the behaviour of the methods on realistic scenarios. The Gradient Projection and the Scaled Gradient Projection algorithms were equipped with Adaptive Barzilai–Borwein steplength selection rule and with recent limited–memory steplength rule based on Ritz–like values; the performances of these methods were compared with each other and with a state–of–the–art method for Computed Tomography problems. Numerical experience showed the effectiveness of the scaling matrix and of the designed steplength selection rules.

Work in progress concerns the investigation of the described methods and acceleration techniques in microwave tomography problems for brain imaging.

Bibliography

[1] D.C. Alexander. *Maximum Entropy Spherical Deconvolution for Diffusion MRI.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[2] A.W. Anderson. Measurement of fiber orientation distributions using high angular resolution diffusion imaging. *Magnetic Resonance in Medicine*, 54(5):1194–1206, 2005.

[3] A. Auría, A. Daducci, J.-P. Thiran, and Y. Wiaux. Structured sparsity for spatially coherent fibre orientation estimation in diffusion MRI. *Neuroimage*, 115:245–255, 2015.

[4] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.

[5] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces.* CMS Books on Mathematics. Springer, 2011.

[6] 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.

[7] 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*, pages 42–88. Cambridge University Press, 2010.

[8] M. Beister, D. Kolditz, and W. Kalender. Iterative reconstruction methods in X-ray CT. *Physica Medica*, 28:94–108, 2012.

[9] F. Benvenuto, R. Zanella, L. Zanni, and M. Bertero. Nonnegative least-squares image deblurring: improved gradient projection approaches. *Inverse Problems*, 26(2), 2010.

[10] M. Bertero and P. Boccacci. *Introduction to inverse problems in imaging.* Institute of Physics Publishing, Bristol, 1998.

[11] 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.

[12] D. Bertsekas. *Nonlinear programming.* Athena Scientific, Belmont, 1999.

[13] D. Bertsekas. *Convex optimization theory. Supplementary Chapter 6 on convex optimization algorithms.* Athena Scientific, Belmont, MA, december 19, 2014 edition, 2009.

[14] 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.

[15] 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, 2003.

[16] 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.

[17] S. Bonettini, F. Porta, and V. Ruggiero. A variable metric inertial method for convex optimization. *SIAM Journal on Scientific Computing*, 31(4):A2558–A2584, 2016.

[18] S. Bonettini and M. Prato. Nonnegative image reconstruction from sparse Fourier data: a new deconvolution algorithm. *Inverse Problems*, 26(9), 2010.

[19] S. Bonettini and M. Prato. New convergence results for the scaled gradient projection method. *Inverse Problems*, 31(9):1196–1211, 2015.

[20] S. Bonettini, R. Zanella, and L. Zanni. A scaled gradient projection method for constrained image deblurring. *Inverse Problems*, 25(1), 2009.

[21] D.J. Brenner and E. Hall. Computed tomography: an increasing source of radiation exposure. *New England Journal of Medicine*, 357:2277–2284, 2007.

[22] T.M. Buzug. *Computed Tomography.* Springer–Verlag, Berlin, Heidelberg, 2008.

[23] E.J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[24] E.J. Candès and M.B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.

[25] E.J. Candès, M.B. Wakin, and S.P. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14:877–905, 2008.

[26] A. Cauchy. Méthode générale pour la résolution des systèmes d'equations simultanées. *Comptes Rendus de l'Académie de Sciences Paris*, 25:536–538, 1847.

[27] 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.

[28] 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, 2015.

[29] 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, 1997.

[30] V.L. Coli, E. Loli Piccolomini, E. Morotti, and L. Zanni. A fast gradient projection method for 3D image reconstruction from limited tomographic data. In *Journal of Physics: Conference Series*, volume 904, page 012013. IOP Publishing, 2017.

[31] V.L. Coli, V. Ruggiero, and L. Zanni. Scaled first-order methods for a class of large-scale constrained least squares problems. In *Numerical Computations: Theory and Algorithms (NUMTA-2016)*, pages 040002–1 – 040002–4. AIP Publishing, 2016.

[32] 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.

[33] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4(4):1168–1200, 2005.

[34] F. E. Curtis and W. Guo. R–Linear Convergence of Limited Memory Steepest Descent. *IMA Journal of Numerical Analysis*, DOI: 10.1093/imanum/drx016, 2017.

[35] A. Daducci, E.J. Canales-Rodríguez, M. Descoteaux, E. Garyfallidis, Y. Gur, Y.-C. Lin, M. Mani, S. Merlet, M. Paquette, A. Ramirez-Manzanares, M. Reisert, P. Reis Rodrigues, F. Sepehrband, E. Caruyer, J. Choupan, R. Deriche, M. Jacob, G. Menegaz, V. Prčkovska, M. Rivera, Y. Wiaux, and J.-P. Thiran. Quantitative comparison of reconstruction methods for intra-voxel fiber recovery from diffusion MRI. *IEEE Transactions on Medical Imaging*, 33(2):384–399, 2014.

[36] A. Daducci, D. Van De Ville, J.-P. Thiran, and Y. Wiaux. Sparse regularization for fiber ODF reconstruction: from the suboptimality of ℓ_2 and ℓ_1 priors to ℓ_0. *Medical Image Analysis*, 18:820–833, 2014.

[37] Y. H. Dai. Alternate step gradient method. *Optimization*, 52:395–415, 2003.

[38] Y. H. Dai and R. Fletcher. Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. *Numerische Mathematik*, 100:21–47, 2005.

[39] 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, 2006.

[40] 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.

[41] Y. H. Dai and Y. X. Yuan. Alternate minimization gradient method. *IMA Journal of Numerical Analysis*, 23(3):377–393, 2003.

[42] M. E. Daube-Witherspoon and G. Muehllener. An iterative image space reconstruction algorithm suitable for volume ECT. *IEEE Transactions on Medical Imaging*, 5(2):61–66, 1986.

[43] 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.

[44] R. De Asmundis, D. di Serafino, W. Hager, G. Toraldo, and H. Zhang. An efficient gradient method using the Yuan steplength. *Computational Optimization and Applications*, 59(3):541–563, 2014.

[45] R. De Asmundis, D. di Serafino, F. Riccio, and G. Toraldo. On spectral properties of steepest descent methods. *IMA J. Numer. Anal.*, 33(4):1416–1435, 2013.

[46] M. Defrise, C. Vanhove, and X. Liu. An algorithm for total variation regularization in high-dimensional linear problems. *Inverse Problems*, 52:329–356, 2011.

[47] F. Dell'acqua, G. Rizzo, P. Scifo, R. Clarke, G. Scotti, and F. Fazio. A model-based deconvolution approach to solve fiber crossing in diffusion-weighted MR imaging. *IEEE Transactions on Biomedical Engineering*, 54:462–472, 2007.

[48] D. di Serafino, V. Ruggiero, G. Toraldo, and L. Zanni. On the steplength selection in gradient methods for unconstrained optimization. *Applied Mathematics and Computation*, 318:176–195, 2018.

[49] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

[50] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of inverse problems.* Kluwer, Dordrecht, 1996.

[51] L. Feldkamp, L. Davis, and J. Kress. Practical cone-beam algorithm. *Journal of the Optical Society of America*, 1:612–619, 1984.

[52] R. Fletcher. Low storage methods for unconstrained optimization. *Lectures in Applied Mathematics*, 26:165–179, 1990.

[53] R. Fletcher. On the Barzilai-Borwein method. *Optimization and Control with Applications*, 96:235–256, 2005.

[54] R. Fletcher. A limited memory steepest descent method. *Mathematical Programming*, 135(1–2):413–436, 2012.

[55] 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, 2015.

[56] 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.

[57] A. Friedlander, J.M. Martinez, B. Molina, and M. Raydan. Gradient method with retards and generalizations. *SIAM Journal on Numerical Analysis*, 36:275–289, 1999.

[58] 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, 1984.

[59] W. Glunt, T.L. Hayden, and M. Raydan. Molecular conformations from distance matrices. *Journal of Computational Chemistry*, 14:114–120, 1993.

[60] G. H. Golub and C. F. Van Loan. *Matrix Computations.* John Hopkins University Press, Baltimore, 3rd edition, 1996.

[61] C. Graff and E. Sidky. Compressive sensing in medical imaging. *Applied Optics*, 54(8):C23–C44, 2015.

[62] 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.

[63] J. Hadamard. *Lectures on Cauchy's problem in linear partial differential equations.* Yale University Press, New Haven, 1923.

[64] K. Hämäläinen, L. Harhanen, A. Hauptmann, A. Kallonen, E. Niemi, and S. Siltanen. Total variation regularization for large-scale X-ray tomography. *International Journal of Tomography and Simulation*, 25(1):1–25, 2014.

[65] P. C. Hansen. *Rank-deficient and discrete ill-posed problems.* SIAM, Philadelphia, 1997.

[66] P. C. Hansen, J. G. Nagy, and D. P. O'Leary. *Deblurring Images: Matrices, Spectra and Filtering.* SIAM, Philadelphia, 2006.

[67] 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, 2012.

[68] A. N. Iusem. On the convergence properties of the projected gradient method for convex optimization. *Computational and Applied Mathematics*, 22(1):37–52, 2003.

[69] T. L. Jensen, J.H. Jørgensen, P.C. Hansen, and S. H. Jensen. Implementation of an optimal first-order method for strongly convex total variation regularization. *BIT Numerical Mathematics*, 52(2):329–356, 2012.

[70] B. Jian and B. Vermuri. A unified computational framework for deconvolution to reconstruct multiple fibers from diffusion weighted mri. *IEEE Transactions on Medical Imaging*, 26:1464–1471, 2007.

[71] J.H. Jørgensen, T.L. Jensen, P.C. Hansen, S.H. Jensen, E.Y. Sidky, and X. Pan. Accelerated gradient methods for total-variation-based CT image reconstruction. In *11th Fully 3D Image Reconstruction in Radiology and Nuclear Medicins*, pages 435–438, 2011.

[72] D. Kim, D. Pal, J. Thibault, and J.A. Fessler. Accelerating ordered subsets image reconstruction for x-ray ct using spatially nonuniform optimization transfer. *IEEE Transactions on Medical Imaging*, 32(11):1965–1978, 2013.

[73] D. Kim, S. Ramani, and J.A. Fessler. Combining ordered subsets and momentum for accelerated X-rays ct imaging reconstruction. *IEEE Transactions on Medical Imaging*, 34(1):167–178, 2015.

[74] K. Lange, D. Hunter, and I. Yang. Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9(1):1–20, 2000.

[75] H. Lantéri, M. Roche, and C. Aime. Penalized maximum likelihood image restoration with positivity constraints: multiplicative algorithms. *Inverse Problems*, 18(5):1397–1419, 2002.

[76] 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, 2001.

[77] E. Loli Piccolomini, V.L. Coli, E. Morotti, and L. Zanni. Reconstruction of 3D X-rays CT images from reduced sampling by a scaled gradient projection algorithm. *Computational Optimization and Applications*, ():1–21, 2017.

[78] E. Loli Piccolomini and E. Morotti. A fast TV-based iterative algorithm for digital breast tomosynthesis image reconstruction. *Journal of Algorithms and Computational Technology*, 10(4):277–289, 2016.

[79] 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, 2009.

[80] J. Mairal. SPAMS: a SPArse Modeling Software, v2.6, 2017.

[81] B. Mani, M. Jacob, A. Guidon, V. Magnotta, and J. Zhong. Acceleration of high angular and spatial resolution diffusion imaging using compressed sensing with multichannel spiral data. *Magnetic Resonance in Medicine*, 73:126–138, 2015.

[82] A. Mastropietro, P. Scifo, and G. Rizzo. Quantitative comparison of spherical deconvolution approaches to resolve complex fiber configurations in diffusion MRI: ISRA-based vs L2L0 sparse methods. *IEEE Transactions on Biomedical Engineering*, 64(12):2847–2857, 2017.

[83] 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.

[84] J.L. Mueller and S. Siltanen. *Linear and Nonlinear Inverse Problems with Practical Applications.* SIAM, Philadelphia, 2012.

[85] H.N. Mülthei. Iterative continuous maximum likelihood reconstruction methods. *Mathematical Methods in the Applied Sciences*, 15:275–286, 1993.

[86] H.N. Mülthei and B. Schorr. On properties of the iterative maximum likelihood reconstruction method. *Mathematical Methods in the Applied Sciences*, 11:331–342, 1989.

[87] 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.

[88] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming Series B*, 140:125–161, 2013.

[89] J. Nocedal and S. J. Wright. *Numerical optimization.* Springer, New York, 2nd edition, 2006.

[90] B.T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4:1–17, 1964.

[91] 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, 2015.

[92] 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, 2012.

[93] A. Ramirez-Manzanares, M. Rivera, B.C. Vemuri, P. Carney, and T. Mareci. Diffusion basis functions decomposition for estimating white matter intravoxel fiber geometry. *IEEE Transactions on Medical Imaging*, 26(8):1091–1102, 2007.

[94] R. Rangayyan, A. Dhawan, and R. Gordon. Algorithms for limited-view computed tomography: an annotated bibliography and a challenge. *Applied Optics*, 24(23):4000–4012, 1985.

[95] M. Raydan. On the Barzilai and Borwein choice of steplength for the gradient method. *IMA Journal of Numerical Analysis*, 13:321–326, 1993.

[96] M. Raydan. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM Journal on Optimization*, 7:26–33, 1997.

[97] M. Raydan and B.F. Svaiter. Relaxed steepest descent and Cauchy-Barzilai-Borwein method. *Computational Optimization and Applications*, 21:155–167, 2002.

[98] R. T. Rockafellar. *Convex analysis.* Princeton University Press, Princeton, NJ, 1970.

[99] R. T. Rockafellar, R. J.-B. Wets, and M. Wets. *Variational Analysis*, volume 317 of *Grundlehren der Mathematischen Wissenschaften.* Springer, Berlin, 1998.

[100] R.T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

[101] S. Rose, M. Andersen, E.Y. Sidky, and X. Pan. Noise properties of CT images reconstructed by use of constrained total-variation, data-discrepancy minimization. *Medical Physics*, 42(5):2690–2698, 2015.

[102] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1–4):259–268, 1992.

[103] L. A. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Transactions on Medical Imaging*, 1(2):113–122, 1982.

[104] R.L. Siddon. Fast calculation of the exact radiological path for a three-dimensional CT array. *Medical Physics*, 12(2):252–255, 1985.

[105] E.Y. Sidky, J.H. Jørgensen, and X. Pan. Convex optimization problem prototyping for image reconstruction in computed tomography with the Chambolle-Pock algorithm. *Physics in Medicine and Biology*, 57(10):3065–3091, 2012.

[106] E.Y. Sidky, J.H. Jørgensen, and X. Pan. First-order convex feasibility for x-ray CT. *Medical Physics*, 40(3):3115–1–15, 2013.

[107] E.Y. Sidky, C.M. Kao, and X. Pan. Accurate image reconstruction from few-views and limited-angle data in divergent-beam CT. *Journal of X-Ray Science and Technology*, 14(2):119–139, 2006.

[108] E.Y. Sidky and X. Pan. Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization. *Physics in Medicine and Biology*, 53:4777–4807, 2008.

[109] E.Y. Sidky, X. Pan, I.S. Reiser, and R.M. Nishikawa. Enhanced imaging of microcalcifications in digital breast tomosynthesis through improved image-reconstruction algorithms. *Medical Physics*, 36(11):4920–4932, 2009.

[110] A.N. Tikhonov. On the stability of inverse problems. *Doklady Akademii Nauk SSSR*, 39(5):195–198, 1943.

[111] A.N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Mathematics Doklady*, 4:1035–1038, 1963.

[112] A.N. Tikhonov and V.Y. Arsenin. *Solution of Ill Posed Problems*. Wiley, New York, 1977.

[113] J.-D. Tournier, F. Calamante, and A. Connelly. Robust determination of the fibre orientation distribution in diffusion MRI: Non-negativity constrained super-resolved spherical deconvolution. *NeuroImage*, 35:1459–1472, 2007.

[114] J.-D. Tournier, F. Calamante, D.G. Gadian, and A. Connelly. Direct estimation of the fiber orientation density function from diffusion-weighted MRI data using spherical deconvolution. *NeuroImage*, 23:1176–1185, 2004.

[115] C. R. Vogel. *Computational methods for inverse problems.* SIAM, Philadelphia, 2002.

[116] H. Yu and G. Wang. A soft-threshold filtering approach for reconstruction from a limited number of projections. *Physics in Medicine and Biology*, 55:3905–3916, 2010.

[117] Y. Yuan. A new stepsize for the steepest descent method. *Journal of Computational Mathematics*, 24:149–156, 2006.

[118] A. Zalinescu. *Convex analysis in general vector spaces.* World Scientific Publishing Co. Inc., River Edge, NJ, 2002.

[119] R. Zanella, P. Boccacci, L. Zanni, and M. Bertero. Efficient gradient projection methods for edge-preserving removal of Poisson noise. *Inverse Problems*, 25(4), 2009.

[120] B. Zhou, L. Gao, and Y. H. Dai. Gradient methods with adaptive step-sizes. *Computational Optimization and Applications*, 35(1):69–86, 2006.