Università degli Studi di Modena
e Reggio Emilia

School of Graduate Studies
Multiscale Modelling, Computational Simulations and Characterization
in Material and Life Sciences
- XXV Cycle -
Director: Prof. Maria Cristina Menziani
Coordinator: Prof. Claudio Giberti

Separable least squares problems and their application to blind deconvolution

Advisor:
Prof. Luca Zanni

Co-advisor:
Dr. Marco Prato

Student:
Anastasia Cornelio

Year: 2013

Contents

Introduction

In any image reconstruction problem, the data at one's disposal are the result of an interaction between the desired image and an acquisition system that transforms the image itself in some kind of available information. One of the most popular examples in this framework is the deblurring problem, in which the measured information is a degraded version of the real image resulting from both the action of the acquisition system point spread function (PSF) and the presence of statistical noise typically due to measurement errors. On the other hand, in many applications the acquisition system hardware encodes the information needed to restore the unknown image in a different space. This happens, for examples, in optical interferometry, where the use of separated telescopes allows to estimate amplitude and phase of the image Fourier components corresponding to some specific fringe spatial frequencies. A similar approach occurs also in several medical applications, as Computerized Tomography or Magnetic Resonance Imaging, in which Fourier samples are achieved by means of the Radon Transform and the Fourier Slice Theorem. Common denominator of these problems is that the relationship between the measured data and the desired image is mathematically given by a Fredholm integral equation of the first kind, in which the kernel function is given by the PSF of the imaging system. A huge amount of literature is available on this topic when the PSF is completely known, especially in the case of a space-invariant system, where the integral relation becomes a convolution operator.

The problem becomes more difficult when the degraded image has to be found without prior knowledge of the system PSF, which is a generally frequent situation in several areas, due to restrictions imposed by the system hardware, high costs or limited time at one's disposal. This problem is typically known as blind deconvolution and, in this case, the problem must be reformulated by introducing as far as possible all available a-priori information on both the object and the PSF. A halfway situation occurs when the system response is not perfectly known, but a parametrized version of the PSF is provided. For this reason, this kind of problem is also called semi-blind or myopic deconvolution (in our thesis, we will use the term "blind" even in this cases). A common strategy

to address the solution of a blind deconvolution problem leads to the minimization of a fit-to-data function, where the unknowns are both the target and a version of the instrumental model, which contains some information about it. The numerical difficulties that characterize this problem are the typical ones of a discrete inverse problem. In fact, the continuous mathematical model is typically a severely ill-posed inverse problem, i.e. the solution might lack in existence, uniqueness or continuous dependence on the data. The ill-posedness can drop once the continuous model is discretized. Unfortunately, the discretization of the problem does not avoid the ill-conditioning pathology, that amplifies the noise affecting the data with the result of a meaningless reconstructed solution. Due to the presence of noise, any naive inversion of the measured data would lead to a numerically unstable solution, making the use of appropriate regularization techniques necessary to recover a physically meaningful reconstructed solution. In this thesis we consider blind deconvolution problems in which the fit-to-data term is given by the classical least-squares functional. In particular, we make the assumption that the blurring operator can be described through a model function depending on some parameters to be estimated. The resulting set of optimization variables is thus naturally partitioned into two groups: the linear variables of the object and the nonlinear parameters describing the blurring operator. For this kind of problems, referred as separable least squares, it's reasonable to perform the minimization over the grouped subsets of variables, in order to exploit the separability by replacing the original joint optimization with sequences of easier subproblems over the partitioned variables. The methods typically used in literature to address separable least squares are Variable Projection and Alternating Optimization.

The Variable Projection method consists of explicitly eliminating a set of variables, in our case the linear one, which depends on the other, in order to obtain a new objective function only on the second nonlinear variables. One obvious advantage is that the iterative nonlinear algorithm used to solve the resulting minimization problem works in a reduced space than the original one. On the other hand, Alternating Optimization iteratively builds up a sequence of subproblems over the two groups of unknowns, which eventually converges to a stationary point of the original problem. In particular, each subproblem restricts the minimization over a set of variables, leaving fixed the other, making Alternating Optimization preferable to joint optimization due to the smaller dimension and, moreover, to the possibility of solving each subproblem inexactly. The main research goal of this thesis concerns the constrained formulation of separable least squares; in particular, due to the nature of the imaging application, our interest is especially dedicated to nonnegative constraints imposed on the object. In order to handle the

nonnegative case, an original formulation of the Variable Projection method has been proposed, together with a new formula for the computation of the Jacobian matrix.

A further contribution is a both theoretical and numerical comparison between Variable Projection and Alternating Optimization for the resolution of blind deconvolution modeled as a separable least squares problem. Since the two approaches seek for the minimization of different objective functions, the key difference lays in the Jacobian matrices of the related residual vectors. For this reason, the Jacobian computation, and its possible approximation, reveals a crucial role and strongly affects the performance of each method. Moreover, the kind of regularization the methods can handle represents a substantial difference between them: indeed, Variable Projection allows just direct regularization, while for Alternating Optimization both direct and iterative regularization are permitted.

The thesis is organized in four chapters. In Chapter 1, the theoretical features of Variable Projection and Alternating Optimization methods are described, focusing on their nonnegatively constrained formulation. By exploiting the closed formula for the nonnegative least squares solution, an original extension of the Variable Projection method to this nonnegative case is proposed. A simple example of Gaussian fitting provides an initial and small-scale comparison of the performance of the methods.

Once introduced the image restoration framework, a regularized descent method for linear least squares problems is proposed in Chapter 2. With the aim of giving special attention to the large-scale aspect, Generalized Cross Validation together with Lanczos bidiagonalization are employed in order to compute a regularized descent direction.

In Chapter 3 blind deconvolution stated as a separable problem has been approached through both Variable Projection and Alternating Optimization. Numerical tests validate the constrained formulation; indeed, significant improvements in the reconstructions are achieved by nonnegative blind deconvolution with respect to the unconstrained case; moreover, numerical results show that, in presence of nonnegative constraints, the Variable Projection objective function behaves fairly better than in the unconstrained case.

In Chapter 4 we approach a real-world application of a blind deconvolution problem, namely the restoration of an image starting from samples of its Fourier Transform, when only partial information about the data frequencies is provided. The uncertainties on the spatial frequencies characterize the problem as a blind deconvolution problem, since the frequencies represent the nonlinear unknowns and allow the formulation as a separable least squares. An application to the imaging of high-energy radiation emitted during a solar flare through the analysis of the photon counts collected by the NASA RHESSI satellite is considered. Since a gradient method has been specifically proposed for solving

the RHESSI deconvolution problem, i.e. when the blurring operator is known, and due to its performed iterative regularization, blind deconvolution is approached only through Alternating Optimization. Also for this application, numerical experiments on simulated data show that the nonnegative blind deconvolution approach provides some improvements in the reconstructions.

Chapter 1

Separable least squares problems

Separable least squares occur when the variables of a, in general, nonlinear least squares problem can be grouped in two or more subsets in order to exploit a particular property of the variables themselves. A typical example occurs when the residual is linear with respect to a subset of components. This particular structure makes the variable separation well suited since it allows to restrict a first subset, where linear optimization techniques can be employed, and a second one of still nonlinear variables, but of smaller dimension than the original problem.

The two most common methods to address separable problems are Variable Projection and Alternating Optimization. A constrained formulation can be handled by both of these approaches; we discuss them in detail, giving special attention to the particular case of nonnegative constraints imposed on the linear variable. An original extension of the Variable Projection method to this nonnegative case and a new formula for the computation of the Jacobian matrix are proposed.

This chapter is organized as follows. In section 1.1 we first outline general properties of least squares; later in section 1.2 we turn to the special case of separable problems. In section 1.3 the Variable Projection method is analyzed, first in the unconstrained and then in the constrained formulation. In 1.3.2 special attention is dedicated to the nonnegative case and a new formula for the computation of the Jacobian matrix is proposed. In section 1.4 we describe the Alternating Optimization approach, both for the constrained and unconstrained case. Finally in section 1.5 we compare the previously described methods, giving a simple numerical experiment.

1.1 Least squares problems

In this section we make a general introduction to least squares problems and briefly discuss their main properties, distinguishing between the linear and nonlinear case. A least squares problem can be stated as

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}) = \|\boldsymbol{r}(\boldsymbol{z})\|_2^2 = \sum_{j=1}^{q} r_j^2(\boldsymbol{z}) \tag{1.1}$$

where $\boldsymbol{r} : \mathbb{R}^p \longrightarrow \mathbb{R}^q$ is the *residual* function and $r_j : \mathbb{R}^p \longrightarrow \mathbb{R}$ is the j-th component. According to the form of the residual function, problem (1.1) has different properties that makes a method more suitable than others to solve it. If $\boldsymbol{r}(\boldsymbol{z})$ is a linear function, then (1.1) becomes a linear least squares problem; on the other hand, when $r(\boldsymbol{z})$ is nonlinear, (1.1) is a nonlinear least squares problem.

A classical example of least squares is parameter identification in data fitting, which consists of describing some data $b_j, j = 1, \ldots, q$ through a nonlinear model function $m(\boldsymbol{z})$ depending on some unknown parameters. In this case, the objective function is the discrepancy between model and data and $r(\boldsymbol{z}) = m(\boldsymbol{z}) - \boldsymbol{b}$. Linear least squares are really common and they also appear often as subproblems in several algorithms, thus in section 1.1.1 we will briefly introduce them before discussing the more general nonlinear case in 1.1.2.

1.1.1 Linear least squares

In a linear least squares problem, the residual is a linear function. If we assume $\boldsymbol{r}(\boldsymbol{z}) = \boldsymbol{A}\boldsymbol{z} - \boldsymbol{b}$, equation (1.1) becomes

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}) = \|\boldsymbol{A}\boldsymbol{z} - \boldsymbol{b}\|_2^2 = \boldsymbol{z}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{z} - 2\boldsymbol{z}^T\boldsymbol{A}^T\boldsymbol{b} + \boldsymbol{b}^T\boldsymbol{b}\,, \tag{1.2}$$

where $\boldsymbol{A} \in \mathbb{R}^{q\times p}$, $\boldsymbol{z} \in \mathbb{R}^p$ and $\boldsymbol{b} \in \mathbb{R}^q$. The gradient and Hessian matrix of f are respectively

$$\nabla f(\boldsymbol{z}) = 2\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{z} - \boldsymbol{b}), \qquad \nabla^2 f(\boldsymbol{z}) = 2\boldsymbol{A}^T\boldsymbol{A}.$$

The objective function $f(\boldsymbol{z})$ in (1.2) is convex. This property is fundamental in unconstrained optimization, since it ensures that any local minimum must be a global minimum and that any stationary point is a global minimum. Thus the solution of the linear least squares problem (1.2) satisfies the so-called *normal equations*

$$\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{z} = \boldsymbol{A}^T\boldsymbol{b}. \tag{1.3}$$

By taking into account that the system matrix $\boldsymbol{A}^T\boldsymbol{A}$ is symmetric, appropriate direct or iterative methods can be used to solve (1.3).

1.1.2 Nonlinear least squares

In this section we discuss the nonlinear least squares problem and describe the different classes of methods to solve them. Given the Jacobian of $\boldsymbol{r}(\boldsymbol{z})$

$$\boldsymbol{J}(\boldsymbol{z}) = \frac{\partial r_j}{\partial z_i} = \begin{bmatrix} \frac{\partial r_1}{\partial z_1} & \cdots & \frac{\partial r_1}{\partial z_p} \\ \vdots & & \vdots \\ \frac{\partial r_q}{\partial z_1} & \cdots & \frac{\partial r_q}{\partial z_p} \end{bmatrix}, \tag{1.4}$$

one can define the gradient of $f(\boldsymbol{z})$ in (1.1) as

$$\nabla f(\boldsymbol{z}) = 2\sum_{j=1}^{q} r_j(\boldsymbol{z})\nabla r_j(\boldsymbol{z}) = 2\boldsymbol{J}(\boldsymbol{z})^T \boldsymbol{r}(\boldsymbol{z}) \tag{1.5}$$

and the Hessian matrix as

$$\begin{aligned} \nabla^2 f(\boldsymbol{z}) =& 2\sum_{j=1}^{q} \nabla r_j(\boldsymbol{z})\nabla r_j(\boldsymbol{z})^T + 2\sum_{j=1}^{q} r_j(\boldsymbol{z})\nabla^2 r_j(\boldsymbol{z}) = \\ =& 2\boldsymbol{J}(\boldsymbol{z})^T \boldsymbol{J}(\boldsymbol{z}) + 2\sum_{j=1}^{q} r_j(\boldsymbol{z})\nabla^2 r_j(\boldsymbol{z})\,. \end{aligned} \tag{1.6}$$

The Jacobian $\boldsymbol{J}$ is a $q \times p$ matrix, whose i-th column is $\dfrac{\partial \boldsymbol{r}}{\partial z_i}$, while the j-th row is ∇r_j. $\boldsymbol{J}$ has a crucial role when solving nonlinear least squares, since by using the Jacobian we can easily calculate the gradient $\nabla f(\boldsymbol{z})$ with (1.5) and an approximation of the Hessian matrix $\nabla^2 f(\boldsymbol{z})$. Indeed, in (1.6) we can see that the Hessian contains two terms:

- $2\boldsymbol{J}^T\boldsymbol{J}$, which involves just the first order derivatives;
- $\boldsymbol{S} = 2\sum_{j=1}^{q} r_j(\boldsymbol{z})\nabla^2 r_j(\boldsymbol{z})$, whose evaluation requires the second order derivatives of the residual.

In the so-called *small residual* case, $\boldsymbol{S}$ is relatively smaller than $2\boldsymbol{J}^T\boldsymbol{J}$. Thus by approximating the Hessian with the first term in (1.6), we keep most of the information, saving computations at the same time. On the other hand, for *large residual* problems the second term may be too significant to be ignored.

Small residual

In the following we describe iterative methods to solve equation (1.1), when the residuals r_j or $\nabla^2 r_j$ are relatively small. In this case we can solve (1.1) by using a Newton-like

method, where the Hessian can be approximated through the only Jacobian matrix. The fundamental difference between the methods concerns the computation of the descent direction. The standard Newton direction is obtained by solving at each iteration k the linear system $\nabla^2 f(\boldsymbol{z}_k)\boldsymbol{d}_k = -\nabla f(\boldsymbol{z}_k)$. By taking advantage of the two additive terms in (1.6), the computation of the q second-order derivatives $\nabla^2 r_j(\boldsymbol{z}_k)$ can be avoided and the search direction can be determined as $2\boldsymbol{J}_k^T\boldsymbol{J}_k\,\boldsymbol{d}_k = -\nabla f(\boldsymbol{z}_k)$. Since, from (1.5), $\nabla f(\boldsymbol{z}_k) = 2\boldsymbol{J}_k^T\boldsymbol{r}_k$, the direction $\boldsymbol{d}_k$ is the solution of the linear least squares problem

$$\min_{\boldsymbol{d}} \|\boldsymbol{J}_k\boldsymbol{d} + \boldsymbol{r}_k\|_2^2 .$$

The resulting method is the Gauss-Newton algorithm:

Gauss-Newton algorithm

choose initial $\boldsymbol{z}_0$

for $k = 0, 1, 2, \ldots$

1. compute the residuals
   $\boldsymbol{r}_k = \boldsymbol{r}(\boldsymbol{z}_k)$
2. compute the Jacobian
   $\boldsymbol{J}_k = \boldsymbol{J}(\boldsymbol{z}_k)$
3. compute the direction by solving
   $2\boldsymbol{J}_k^T\boldsymbol{J}_k\,\boldsymbol{d}_k = -2\boldsymbol{J}_k^T\boldsymbol{r}_k$
4. update the solution
   $\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \boldsymbol{d}_k$

end

The Gauss-Newton algorithm is *locally* convergent [38, 43]; for this reason a line search in the direction $\boldsymbol{d}_k$ is usually performed to ensure global convergence. The solution is then updated as

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \tau_k\boldsymbol{d}_k ,$$

where the steplength τ_k has to satisfy the Wolfe conditions.

An alternative to Gauss-Newton in the small residual case is the Levenberg-Marquardt algorithm, which can be considered as a modification of the Gauss-Newton method. Indeed it uses a similar Hessian approximation, involving only the 1-st order derivatives,

but requires a parameter $\lambda \geq 0$ for the computation of the direction: $\boldsymbol{d}_k$ is the solution of the linear system

$$(2\boldsymbol{J}_k^T \boldsymbol{J}_k + 2\lambda \boldsymbol{I})\boldsymbol{d}_k = -2\boldsymbol{J}_k^T \boldsymbol{r}_k$$

or, equivalently, as the solution of the linear least squares

$$\min_{\boldsymbol{d}} \left\| (\boldsymbol{J}_k + \sqrt{\lambda})\boldsymbol{d} + \boldsymbol{r}_k \right\|_2^2 .$$

Adding a positive constant to the diagonal of $2\boldsymbol{J}_k^T \boldsymbol{J}_k$ is really convenient when the Jacobian is rank-deficient or ill-conditioned. In [43] Nocedal and Wright analyze the method using the trust-region framework.

Since similar Hessian approximations are used in each case, the local convergence properties of Levenberg-Marquardt are similar to those of Gauss-Newton.

1.2 Separable least squares

In this section we investigate nonlinear least squares problems of the form

$$\min_{\boldsymbol{z} \in \mathbb{R}^p} f(\boldsymbol{z}) = \|F(\boldsymbol{z})\|_2^2 \tag{1.7}$$

whose unknowns can be partitioned into two subsets, so that

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix}, \quad \boldsymbol{x} \in \mathbb{R}^n, \quad \boldsymbol{y} \in \mathbb{R}^m \quad \text{and} \quad n + m = p$$

where we can assume $m \ll n$. Equation (1.7) thus becomes

$$\min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y}) = \|F(\boldsymbol{x}, \boldsymbol{y})\|_2^2 \tag{1.8}$$

and in this work will be referred to as a *separable nonlinear least squares* problem. Every kind of variable separation is possible, but it's reasonable as well as desirable to exploit the separability in order to simplify the original problem (1.7). The most useful situations for variable separation occur when

1. F **is linear with respect to** $\boldsymbol{x}$. In this case the residual F in (1.7) becomes

   $$F(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{A}(\boldsymbol{y})\boldsymbol{x} - \boldsymbol{b}$$

   where $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^m$, $\boldsymbol{b} \in \mathbb{R}^q$ and $\boldsymbol{A} : \mathbb{R}^m \to \mathbb{R}^{q \times n}$. By exploiting the separability, (1.7) becomes a linear least squares problem with respect to $\boldsymbol{x}$.

2. **Both $\boldsymbol{x}$ and $\boldsymbol{y}$ occur linearly on** F so that (1.7) can be written as

$$\min_{\boldsymbol{x},\boldsymbol{y}} \|\boldsymbol{y}\boldsymbol{x}^T - \boldsymbol{B}\|_2^2$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^m$ and $\boldsymbol{B} \in \mathbb{R}^{m \times n}$.

We will focus on applications belonging to the first kind of problems, widely treated in literature in two different approaches, which are *Variable Projection* and *Alternating Optimization*.

The ***Variable Projection*** method has been proposed by Golub and Pereyra in 1973 [26] and it is based on the explicit elimination of the $\boldsymbol{x}$ variable in order to obtain a new function depending only on the other variable $\boldsymbol{y}$. In particular, for every fixed $\boldsymbol{y}$, the subproblem

$$\boldsymbol{x}(\boldsymbol{y}) = \operatorname*{argmin}_{\boldsymbol{x}} \|F(\boldsymbol{x}, \boldsymbol{y})\|_2^2 \tag{1.9}$$

becomes a linear least squares and $\boldsymbol{x}(\boldsymbol{y})$ can be inserted into the original problem (1.7) obtaining a new reduced minimization problem depending only on $\boldsymbol{y}$:

$$\min_{\boldsymbol{y} \in \mathbb{R}^m} \varphi(\boldsymbol{y}) = f(\boldsymbol{x}(\boldsymbol{y}), \boldsymbol{y})\,. \tag{1.10}$$

On the other hand is the ***Alternating Optimization*** method, widely investigated in literature from different research groups. In this work we will follow the approach proposed by Bedzek et al. in 1987 in [9], at first referred to as *Grouped Coordinate Minimization*. The method defines an iterative sequence $\{(\boldsymbol{x}_{\ell+1}, \boldsymbol{y}_{\ell+1})\} \in \mathbb{R}^n \times \mathbb{R}^m$ where

$$\begin{aligned} \boldsymbol{x}_{\ell+1} &= \operatorname*{argmin}_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{y}_\ell) \\ \boldsymbol{y}_{\ell+1} &= \operatorname*{argmin}_{\boldsymbol{y}} f(\boldsymbol{x}_{\ell+1}, \boldsymbol{y}) \end{aligned}$$

that, under certain conditions, convergences to a local minimizer $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ of f.

In the rest of this chapter we will describe in detail the two methods, pointing out the main differences between them, their related pros and cons and for which applications it is preferable to apply one method rather than the other.

1.3 Variable Projection

The Variable Projection method has been proposed by Golub and Pereyra [26] in the case where $\boldsymbol{x}$ is a linear subset of variables, but its basic idea can be extended to any arbitrary variable partitioning. The main contributions given in [26] are:

- the proof that the original and the separated problems have the same solutions;
- the differentiation of the pseudoinverse matrix, required for the evaluation of the gradient of the objective function and, consequently, of the Jacobian matrix.

A simplified version of the Jacobian has been proposed in [36] by Kaufman, who observed that in the small residual case a term of the Jacobian can be neglected. Moreover, Kaufman and Pereyra dealt with a constrained formulation of separable nonlinear least squares problems in [37]. They showed that, when equality constraints are imposed, one can reduce the problem to an unconstrained separable nonlinear least squares problem that can be solved with the Variable Projection method. Several implementations of this method have been developed during the years. Extensions to the general nonlinear case of two arbitrary sets of variables are due to Ruhe and Wedin [50].

In this work we will describe separable nonlinear least squares problems where the parameters can be divided into two sets: a set of nonlinear variables $\boldsymbol{y}$ and a set of linear variables $\boldsymbol{x}$, which are dependent on the other. For instance, you can imagine a nonlinear data fitting problem, where the model can be written as a linear combination of nonlinear functions.

Separable nonlinear least squares problems can be stated as

$$\min_{\boldsymbol{x},\boldsymbol{y}} \varphi_0(\boldsymbol{x},\boldsymbol{y}) = \|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x} - \boldsymbol{b}\|_2^2, \tag{1.11}$$

where $\boldsymbol{b} \in \mathbb{R}^q$ is a measured noisy data vector, $\boldsymbol{y} \in \mathbb{R}^m$ is a vector of *nonlinear* parameters, $\boldsymbol{x} \in \mathbb{R}^n$ is a vector of *linear* variables and $\boldsymbol{A}$ is a nonlinear operator that maps $\boldsymbol{y}$ into a $q \times n$ matrix.

The variable projection method [26, 25, 36, 37, 44, 50, 52] applied to (1.11) consists of projecting the linear variables $\boldsymbol{x}$ and obtaining a reduced cost functional depending only on $\boldsymbol{y}$. In particular, if $\boldsymbol{x}$ is determined by solving the linear least squares problem

$$\boldsymbol{x}^{ls}(\boldsymbol{y}) = \operatorname*{argmin}_x \|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x} - \boldsymbol{b}\|_2 \tag{1.12}$$

then (1.12) can be inserted into (1.11) obtaining

$$\min_{\boldsymbol{y}} \varphi(\boldsymbol{y}) = \varphi_0(\boldsymbol{x}^{ls}(\boldsymbol{y}), \boldsymbol{y})\,. \tag{1.13}$$

The linear least squares solution (1.12) can be written through a closed formula as

$$\boldsymbol{x}^{ls}(\boldsymbol{y}) = \boldsymbol{A}(\boldsymbol{y})^\dagger \boldsymbol{b}\,, \tag{1.14}$$

where the symbol † denotes the pseudoinverse matrix. By replacing (1.14) in (1.13), the objective function becomes

$$\varphi(\boldsymbol{y}) = \|\boldsymbol{r}_\varphi(\boldsymbol{y})\|_2^2 = \|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{A}(\boldsymbol{y})^\dagger \boldsymbol{b} - \boldsymbol{b}\|_2^2\,, \tag{1.15}$$

where $\boldsymbol{r}_\varphi(\boldsymbol{y}) = \boldsymbol{A}(\boldsymbol{y})\boldsymbol{A}(\boldsymbol{y})^\dagger \boldsymbol{b} - \boldsymbol{b}$ is the residual function of (1.15). The minimization of (1.15) can be solved with classical nonlinear least squares methods, such as Gauss-Newton or Levenberg-Marquardt algorithm.

Gauss-Newton Algorithm to solve: $\min_{\boldsymbol{y}} \varphi(\boldsymbol{y})$

choose initial $\boldsymbol{y}_0$

for $\ell = 0, 1, 2, \ldots$

$\boldsymbol{x}_{\ell+1} = \boldsymbol{A}(\boldsymbol{y}_\ell)^\dagger \boldsymbol{b}$

$\boldsymbol{r}_\ell = \boldsymbol{A}(\boldsymbol{y}_\ell)\boldsymbol{x}_{\ell+1} - \boldsymbol{b}$

$\boldsymbol{d}_\ell = \operatorname{argmin}_{\boldsymbol{d}} \|\boldsymbol{J}_\ell \boldsymbol{d} + \boldsymbol{r}_\ell\|_2$, where $\boldsymbol{J}_\ell$ is the Jacobian of $\boldsymbol{r}_\varphi$

determine steplength τ_ℓ

$\boldsymbol{y}_{\ell+1} = \boldsymbol{y}_\ell + \tau_\ell \boldsymbol{d}_\ell$

end

The main advantages of this approach are that (1.12) can be solved by a linear method and that the remaining nonlinear problem (1.13) is of smaller dimension than the original (1.11). On the other hand every evaluation of the residual vector or of the objective function requires the solution of the n-dimensional problem in (1.12).

In [26] Golub and Pereyra proved that the problems in (1.11) and (1.13) have the same solutions. More precisely, they stated the following

Theorem 1. Let $\boldsymbol{A}(\boldsymbol{y})$ have a locally constant rank. Then

- if $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is a global minimizer of $\varphi_0(\boldsymbol{x}, \boldsymbol{y})$, then $\boldsymbol{y}^*$ is a global minimizer of $\varphi(\boldsymbol{y})$ and $\varphi(\boldsymbol{y}^*) = \varphi_0(\boldsymbol{x}^*, \boldsymbol{y}^*)$.
- if $\boldsymbol{y}^*$ is a critical point (or global minimizer) of $\varphi(\boldsymbol{y})$ and $\boldsymbol{x}^* = \boldsymbol{A}(\boldsymbol{y}^*)^\dagger \boldsymbol{b}$, then $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is a critical point (or global minimizer) of $\varphi_0(\boldsymbol{x}, \boldsymbol{y})$ and $\varphi_0(\boldsymbol{x}^*, \boldsymbol{y}^*) = \varphi(\boldsymbol{y}^*)$.

The proof of Theorem 1 requires to compute the gradient $\varphi(\boldsymbol{y})$, which involves the Jacobian of the residual function and, consequently, differentiation of the pseudoinverse matrix.

Computing the Jacobian

From [26] we know that the differentiation of the pseudoinverse is

$$\begin{aligned}\frac{\partial \boldsymbol{A}^\dagger}{\partial \boldsymbol{y}} =& -(\boldsymbol{A}^T\boldsymbol{A})^{-1}\left(\frac{\partial \boldsymbol{A}^T}{\partial \boldsymbol{y}}\boldsymbol{A}+\boldsymbol{A}^T\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{y}}\right)\boldsymbol{A}^\dagger+(\boldsymbol{A}^T\boldsymbol{A})^{-1}\frac{\partial \boldsymbol{A}^T}{\partial \boldsymbol{y}}= \\ =&(\boldsymbol{A}^T\boldsymbol{A})^{-1}\frac{\partial \boldsymbol{A}^T}{\partial \boldsymbol{y}}\left(\boldsymbol{I}-\boldsymbol{A}\boldsymbol{A}^\dagger\right)-\boldsymbol{A}^\dagger\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{y}}\boldsymbol{A}^\dagger\end{aligned} \tag{1.16}$$

Thus the k-th column of the Jacobian matrix $\boldsymbol{J}_\varphi$ can be written as [26]

$$\begin{aligned}(\boldsymbol{J}_\varphi)_k =&\frac{\partial \boldsymbol{r}_\varphi}{\partial y_k}=\frac{\partial \boldsymbol{A}}{\partial y_k}\boldsymbol{A}^\dagger\boldsymbol{b}+\boldsymbol{A}\frac{\partial \boldsymbol{A}^\dagger}{\partial y_k}\boldsymbol{b} \\ =&\frac{\partial \boldsymbol{A}}{\partial y_k}\boldsymbol{A}^\dagger\boldsymbol{b}+\boldsymbol{A}\left((\boldsymbol{A}^T\boldsymbol{A})^{-1}\frac{\partial \boldsymbol{A}^T}{\partial y_k}\left(\boldsymbol{I}-\boldsymbol{A}\boldsymbol{A}^\dagger\right)-\boldsymbol{A}^\dagger\frac{\partial \boldsymbol{A}}{\partial y_k}\boldsymbol{A}^\dagger\right)\boldsymbol{b} \\ =&\left(\boldsymbol{I}-\boldsymbol{A}\boldsymbol{A}^\dagger\right)\frac{\partial \boldsymbol{A}}{\partial y_k}\boldsymbol{A}^\dagger\boldsymbol{b}+\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{A})^{-1}\frac{\partial \boldsymbol{A}^T}{\partial y_k}\left(\boldsymbol{I}-\boldsymbol{A}\boldsymbol{A}^\dagger\right)\boldsymbol{b}\end{aligned} \tag{1.17}$$

In [36] Kaufman proposed an algorithm for separable nonlinear least squares problems, with a simplified version of the Jacobian, $\widetilde{J_\varphi}(\boldsymbol{y})$, here referred to as *Kaufman's simplification*. It can be obtained by ignoring the second term of (1.17), which means that its k-th column can be written as:

$$(\widetilde{\boldsymbol{J}_\varphi})_k=\left(\boldsymbol{I}-\boldsymbol{A}\boldsymbol{A}^\dagger\right)\frac{\partial \boldsymbol{A}}{\partial y_k}\boldsymbol{A}^\dagger\boldsymbol{b}. \tag{1.18}$$

1.3.1 Constrained variable projection

We are interested in an extension of the variable projection method in the case of simple constraints imposed on the linear and/or nonlinear variables. How do the constraints modify the method?

Constraints on the nonlinear variable $\boldsymbol{y}$ do not affect the computation of $\boldsymbol{x}$ through (1.12). However, the classical Gauss-Newton or Levenberg-Marquardt algorithms have to be replaced by an opportune constrained method.

Constraints on the linear variable $\boldsymbol{x}$ affect widely the computation of the Jacobian of the objective function and linear projection methods must be applied in order to determine $\boldsymbol{x}$.

However, both kinds of constraints save the separability of the variables, obtaining an outer nonlinear (constrained) optimization problem in $\boldsymbol{y}$ and an inner linear (constrained) least squares problem in $\boldsymbol{x}$.

In [37] Kaufman and Pereyra proposed a method for separable nonlinear least squares problems for separable equality constraints. They reduced the original problem to an unconstrained separable problem, for which Theorem 1 still holds.

In the following we will do the same for the particular case of nonnegative constraints on the linear variable.

1.3.2 A special case: nonnegative constraints on $\boldsymbol{x}$

When nonnegative constraints are imposed on $\boldsymbol{x}$, the original separable problem is

$$\min_{\boldsymbol{x},\boldsymbol{y}} \psi_0(\boldsymbol{x},\boldsymbol{y}) = \|\boldsymbol{r}_\psi(\boldsymbol{y})\|_2^2 \quad \text{subject to } \boldsymbol{x} \geq \boldsymbol{0}\,, \tag{1.19}$$

that, once the variable has been projected, becomes

$$\min_{\boldsymbol{y}} \psi(\boldsymbol{y}) = \psi_0(\boldsymbol{x}^*(\boldsymbol{y}),\boldsymbol{y})\,, \tag{1.20}$$

where $\boldsymbol{x}^*(\boldsymbol{y})$ is the nonnegative least squares solution, that is

$$\boldsymbol{x}^*(\boldsymbol{y}) = \operatorname*{argmin}_{\boldsymbol{x}\geq 0} \|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x} - \boldsymbol{b}\|_2\,. \tag{1.21}$$

Once defined the residual vector $\boldsymbol{r}_\psi(\boldsymbol{y}) = \boldsymbol{A}(\boldsymbol{y})\boldsymbol{x}^*(\boldsymbol{y}) - \boldsymbol{b}$, problem (1.20) can be solved with a classical nonlinear least squares method, such as the Gauss-Newton algorithm [43], here rewritten as:

Gauss-Newton Algorithm to solve: $\min_{\boldsymbol{y}} \psi(\boldsymbol{y}) = \psi_0(\boldsymbol{x}^*(\boldsymbol{y}),\boldsymbol{y})$

choose initial $\boldsymbol{y}_0$

for $\ell = 0, 1, 2, \ldots$

Compute the gradient $\boldsymbol{g}_\ell = \nabla_{\boldsymbol{y}}\psi(\boldsymbol{y}_\ell)$

Compute the Jacobian $\boldsymbol{J}_\ell$ of $\boldsymbol{r}_\psi$

$\boldsymbol{d}_\ell = -(2\boldsymbol{J}_\ell^T\boldsymbol{J}_\ell)^{-1}\boldsymbol{g}_\ell$

determine steplength τ_ℓ

$\boldsymbol{y}_{\ell+1} = \boldsymbol{y}_\ell + \tau_\ell \boldsymbol{d}_\ell$

end

In the presence of constraints on the linear variable $\boldsymbol{x}$, the evaluation of the gradient of $\psi(\boldsymbol{y})$ and of the Jacobian matrix of $\boldsymbol{r}_\psi(\boldsymbol{y})$ requires further discussion. We proceed step by step, starting with the closed formula for the nonnegative least squares solution, then introducing the residual vector and finally obtaining the Jacobian and the gradient.

Closed formula for nonnegative least squares

Like in any nonnegatively constrained optimization problem, for any $\boldsymbol{x} \geq 0$ we can define:

- the active set by $\mathcal{A}(\boldsymbol{x}) = \{i \,|\, x_i = 0\}$;
- and a diagonal matrix $\boldsymbol{D}(\boldsymbol{x})$ by

$$\boldsymbol{D}(\boldsymbol{x})_{ii} = \begin{cases} 1 & \text{if} \quad i \notin \mathcal{A}(\boldsymbol{x}) \\ 0 & \text{if} \quad i \in \mathcal{A}(\boldsymbol{x}) \end{cases} . \tag{1.22}$$

In the following we will use the notation $\boldsymbol{D}^* = \boldsymbol{D}(\boldsymbol{x}^*(\boldsymbol{y}))$.

Therefore the solution of (1.21) can be written by a closed formula as in [4]:

$$\boldsymbol{x}^*(\boldsymbol{y}) = \big(\boldsymbol{A}(\boldsymbol{y})\boldsymbol{D}^*\big)^\dagger \boldsymbol{b}\,. \tag{1.23}$$

Let us briefly comment on the computation of the pseudoinverse $(\boldsymbol{A}\boldsymbol{D}^*)^\dagger$. We may suppose that the active set contains $n - r$ elements and, without loss of generality, that

$$\boldsymbol{D}^* = \begin{bmatrix} \boldsymbol{I}_r & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} .$$

If we partition the first r and the remaining $n - r$ columns of $\boldsymbol{A}$, that is $\boldsymbol{A} = [\,\boldsymbol{A}_1 \;\; \boldsymbol{A}_2\,]$, then $\boldsymbol{A}\boldsymbol{D} = [\,\boldsymbol{A}_1 \;\; \boldsymbol{0}]$. Now, the pseudoinverse can be defined as follows [4]

$$(\boldsymbol{A}\boldsymbol{D}^*)^\dagger = \begin{bmatrix} \boldsymbol{A}_1^\dagger \\ \boldsymbol{0} \end{bmatrix} .$$

Evaluate the residual vector

In least squares problems, as (1.20), the residual vector plays a crucial role, since its definition allows the evaluation of the objective function, the Jacobian matrix and the gradient vector. The advantage of having the solution (1.21) in the closed formula (1.23) is that now it can be explicitly inserted into (1.20) to obtain

$$\min_{\boldsymbol{y}} \psi(\boldsymbol{y}) = \Big\| \boldsymbol{A}(\boldsymbol{y})\big(\boldsymbol{A}(\boldsymbol{y})\boldsymbol{D}^*\big)^\dagger \boldsymbol{b} - \boldsymbol{b} \Big\|_2^2 . \tag{1.24}$$

The previous equation allows to define the *residual vector* $\boldsymbol{r}\colon \mathbb{R}^m \longrightarrow \mathbb{R}^n$ as

$$\boldsymbol{r}(\boldsymbol{y}) = \boldsymbol{A}(\boldsymbol{y})\big(\boldsymbol{A}(\boldsymbol{y})\boldsymbol{D}^*\big)^\dagger \boldsymbol{b} - \boldsymbol{b} \tag{1.25}$$

and consequently the objective function as $\psi(\boldsymbol{y}) = \|\boldsymbol{r}(\boldsymbol{y})\|_2^2$.

At this point, the role of $\boldsymbol{D}^*$ needs some comment. As we can see in (1.23), $\boldsymbol{x}^*(\boldsymbol{y})$ depends on $\boldsymbol{D}^*$; but in the meanwhile, $\boldsymbol{D}^* = \boldsymbol{D}(\boldsymbol{x}^*(\boldsymbol{y}))$ thus $\boldsymbol{D}^*$ depends on $\boldsymbol{x}^*(\boldsymbol{y})$ itself. In order to avoid this concept to be confusing, we discuss it more in details. The expression of the nonnegative solution through the closed formula (1.23) implicitly depends on $\boldsymbol{x}^*(\boldsymbol{y})$. One might ask: is it still useful? And for what? The answer to these reasonable questions is that the closed formula (1.23) is fundamental in order to define ψ as depending only on $\boldsymbol{y}$ and, as we will see later, in order to compute the Jacobian. But, actually, it can't be used to compute the solution and must be replaced by the iterative solution of $\min_{\boldsymbol{x}\geq\boldsymbol{0}} \|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x} - \boldsymbol{b}\|_2^2$.

From this discussion it follows that, for the evaluation of the residual (1.25) at each Gauss-Newton iteration $\ell = 0, 1, 2, \ldots$, we have

$$\begin{aligned} \boldsymbol{x}_{\ell+1} &= \operatorname*{argmin}_{\boldsymbol{x}\geq\boldsymbol{0}} \|\boldsymbol{A}(\boldsymbol{y}_\ell)\boldsymbol{x} - \boldsymbol{b}\|_2 \\ \boldsymbol{r}_\ell &= \boldsymbol{A}(\boldsymbol{y}_\ell)\boldsymbol{x}_{\ell+1} - \boldsymbol{b}\,, \end{aligned}$$

that is, we need to previously perform a projection method to obtain the nonnegative solution (1.21). An approximation of $\boldsymbol{D}^*$ can be obtained through a convergence property of the projection methods. In [38] and [7] the authors prove that the gradient and the Newton projection methods, respectively, identify the binding constraint at the solution $\boldsymbol{x}^*$ in a finite number of iterations. That is, there exists a j^* such that the active set $\mathcal{A}(\boldsymbol{x}^*) = \mathcal{A}(\boldsymbol{x}_j)$ for each $j > j^*$. Since $\boldsymbol{D}$ is defined by the active set, this ensures that

$$\boldsymbol{D}(\boldsymbol{x}^*) = \boldsymbol{D}(\boldsymbol{x}_j) \qquad \text{for each } j > j^*.$$

This property is definitely useful, as we will see later, in the computation of the Jacobian, where $\boldsymbol{D}^*$ occurs. Indeed, during the Gauss-Newton iteration ℓ, once the projection method is used to compute $\boldsymbol{x}_{\ell+1}$, one can determine $\boldsymbol{D}^*$ as $\boldsymbol{D}(\boldsymbol{x}_{\ell+1})$.

Computing the Jacobian

In this section we compute the Jacobian matrix of the residual vector (1.25). In order to simplify the notation, let us first denote $\boldsymbol{A}\boldsymbol{D}^*$ with $\boldsymbol{B}$. Since the residual is $\boldsymbol{r}_\psi(\boldsymbol{y}) = \boldsymbol{A}(\boldsymbol{y})\boldsymbol{B}(\boldsymbol{y})^\dagger \boldsymbol{b} - \boldsymbol{b}$, the k-th column of the Jacobian matrix can be thus obtained

by vectorizing $(\boldsymbol{J}_\psi)_k$, which has the following expression:

$$\begin{aligned}(\boldsymbol{J}_\psi)_k &= \frac{\partial \boldsymbol{r}_\psi}{\partial y_k} = \frac{\partial \boldsymbol{A}}{\partial y_k}\boldsymbol{B}^\dagger \boldsymbol{b} + \boldsymbol{A}\frac{\partial \boldsymbol{B}^\dagger}{\partial y_k}\boldsymbol{b} \\ &= \frac{\partial \boldsymbol{A}}{\partial y_k}\boldsymbol{B}^\dagger \boldsymbol{b} + \boldsymbol{A}\left((\boldsymbol{B}^T\boldsymbol{B})^{-1}\frac{\partial \boldsymbol{B}^T}{\partial y_k}\left(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{B}^\dagger\right) - \boldsymbol{B}^\dagger\frac{\partial \boldsymbol{B}}{\partial y_k}\boldsymbol{B}^\dagger\right)\boldsymbol{b} \end{aligned} \tag{1.26}$$

Now we replace $\boldsymbol{B}$ with $\boldsymbol{A}\boldsymbol{D}^*$; furthermore we recall that $(\boldsymbol{A}\boldsymbol{D}^*)^\dagger \boldsymbol{b} = \boldsymbol{x}^*$ and that

$$\frac{\partial(\boldsymbol{A}\boldsymbol{D}^*)}{\partial y_k} = \left[\frac{\partial \boldsymbol{A}}{\partial y_k}\boldsymbol{D}^* + \boldsymbol{A}\frac{\partial \boldsymbol{D}^*}{\partial y_k}\right].$$

Equation (1.26) can be thus continued as

$$\begin{aligned}(\boldsymbol{J}_\psi)_k =& \frac{\partial \boldsymbol{A}}{\partial y_k}(\boldsymbol{A}\boldsymbol{D}^*)^\dagger \boldsymbol{b} \\ &+ \boldsymbol{A}\left((\boldsymbol{A}\boldsymbol{D}^*)^T(\boldsymbol{A}\boldsymbol{D}^*)\right)^{-1}\frac{\partial(\boldsymbol{A}\boldsymbol{D}^*)^T}{\partial y_k}\left(\boldsymbol{I} - (\boldsymbol{A}\boldsymbol{D}^*)(\boldsymbol{A}\boldsymbol{D}^*)^\dagger\right)\boldsymbol{b} \\ &- \boldsymbol{A}(\boldsymbol{A}\boldsymbol{D}^*)^\dagger\frac{\partial(\boldsymbol{A}\boldsymbol{D}^*)}{\partial y_k}(\boldsymbol{A}\boldsymbol{D}^*)^\dagger \boldsymbol{b} \\ =& \frac{\partial \boldsymbol{A}}{\partial y_k}(\boldsymbol{A}\boldsymbol{D}^*)^\dagger \boldsymbol{b} \\ &+ \boldsymbol{A}\left((\boldsymbol{A}\boldsymbol{D}^*)^T(\boldsymbol{A}\boldsymbol{D}^*)\right)^{-1}\left[\frac{\partial \boldsymbol{A}}{\partial y_k}\boldsymbol{D}^* + \boldsymbol{A}\frac{\partial \boldsymbol{D}^*}{\partial y_k}\right]^T\left(\boldsymbol{I} - (\boldsymbol{A}\boldsymbol{D}^*)(\boldsymbol{A}\boldsymbol{D}^*)^\dagger\right)\boldsymbol{b} \\ &- \boldsymbol{A}(\boldsymbol{A}\boldsymbol{D}^*)^\dagger\left[\frac{\partial \boldsymbol{A}}{\partial y_k}\boldsymbol{D}^* + \boldsymbol{A}\frac{\partial \boldsymbol{D}^*}{\partial y_k}\right](\boldsymbol{A}\boldsymbol{D}^*)^\dagger \boldsymbol{b} \\ =& \frac{\partial \boldsymbol{A}}{\partial y_k}(\boldsymbol{A}\boldsymbol{D}^*)^\dagger \boldsymbol{b} \\ &+ \boldsymbol{A}\left((\boldsymbol{A}\boldsymbol{D}^*)^T(\boldsymbol{A}\boldsymbol{D}^*)\right)^{-1}\left[\frac{\partial \boldsymbol{A}}{\partial y_k}\boldsymbol{D}^*\right]^T\left(\boldsymbol{I} - (\boldsymbol{A}\boldsymbol{D}^*)(\boldsymbol{A}\boldsymbol{D}^*)^\dagger\right)\boldsymbol{b} \\ &+ \boldsymbol{A}\left((\boldsymbol{A}\boldsymbol{D}^*)^T(\boldsymbol{A}\boldsymbol{D}^*)\right)^{-1}\left[\boldsymbol{A}\frac{\partial \boldsymbol{D}^*}{\partial y_k}\right]^T\left(\boldsymbol{I} - (\boldsymbol{A}\boldsymbol{D}^*)(\boldsymbol{A}\boldsymbol{D}^*)^\dagger\right)\boldsymbol{b} \\ &- \boldsymbol{A}(\boldsymbol{A}\boldsymbol{D}^*)^\dagger\left[\frac{\partial \boldsymbol{A}}{\partial y_k}\boldsymbol{D}^*\right](\boldsymbol{A}\boldsymbol{D}^*)^\dagger \boldsymbol{b} - \boldsymbol{A}(\boldsymbol{A}\boldsymbol{D}^*)^\dagger\left[\boldsymbol{A}\frac{\partial \boldsymbol{D}^*}{\partial y_k}\right](\boldsymbol{A}\boldsymbol{D}^*)^\dagger \boldsymbol{b} \\ =& \left(\boldsymbol{I} - \boldsymbol{A}(\boldsymbol{A}\boldsymbol{D}^*)^\dagger\right)\frac{\partial \boldsymbol{A}}{\partial y_k}(\boldsymbol{A}\boldsymbol{D}^*)^\dagger \boldsymbol{b} \\ &+ \boldsymbol{A}\left((\boldsymbol{A}\boldsymbol{D}^*)^T(\boldsymbol{A}\boldsymbol{D}^*)\right)^{-1}\left[\frac{\partial \boldsymbol{A}}{\partial y_k}\boldsymbol{D}^*\right]^T\left(\boldsymbol{I} - (\boldsymbol{A}\boldsymbol{D}^*)(\boldsymbol{A}\boldsymbol{D}^*)^\dagger\right)\boldsymbol{b} \\ &+ \boldsymbol{A}\left((\boldsymbol{A}\boldsymbol{D}^*)^T(\boldsymbol{A}\boldsymbol{D}^*)\right)^{-1}\left[\boldsymbol{A}\frac{\partial \boldsymbol{D}^*}{\partial y_k}\right]^T\left(\boldsymbol{I} - (\boldsymbol{A}\boldsymbol{D}^*)(\boldsymbol{A}\boldsymbol{D}^*)^\dagger\right)\boldsymbol{b} \\ &- \boldsymbol{A}(\boldsymbol{A}\boldsymbol{D}^*)^\dagger\left[\boldsymbol{A}\frac{\partial \boldsymbol{D}^*}{\partial y_k}\right](\boldsymbol{A}\boldsymbol{D}^*)^\dagger \boldsymbol{b}. \end{aligned} \tag{1.27}$$

We observe that $\boldsymbol{D}^*$ is not continuously differentiable with respect to $\boldsymbol{y}$. Hence the term $\dfrac{\partial \boldsymbol{D}^*}{\partial y_k}$ in (1.27) can only be approximated with finite differences.

Computing the gradient

Like in any least squares problem, the gradient vector of ψ is $\boldsymbol{g}(\boldsymbol{y}) = 2\boldsymbol{J}^T\boldsymbol{r}(\boldsymbol{y})$; therefore, at iteration ℓ of Gauss-Newton, the gradient $\boldsymbol{g}_\ell$ and the descent direction $\boldsymbol{d}_\ell$ are respectively

$$\begin{aligned}\boldsymbol{g}_\ell &= 2\boldsymbol{J}_\ell^T\boldsymbol{r}_\ell \\ \boldsymbol{d}_\ell &= -(\boldsymbol{J}_\ell^T\boldsymbol{J}_\ell)^{-1}\boldsymbol{J}_\ell^T\boldsymbol{r}_\ell = \underset{\boldsymbol{d}}{\operatorname{argmin}} \|\boldsymbol{J}_\ell\boldsymbol{d} + \boldsymbol{r}_\ell\|_2\end{aligned}$$

After all these clarifications, we rewrite now the Gauss-Newton algorithm:

Gauss-Newton Algorithm to solve: $\min_{\boldsymbol{y}} \psi(\boldsymbol{y})$

choose initial $\boldsymbol{y}_0$

for $\ell = 0, 1, 2, \ldots$

$\boldsymbol{x}_{\ell+1} = \underset{\boldsymbol{x}\geq \boldsymbol{0}}{\operatorname{argmin}} \|\boldsymbol{A}(\boldsymbol{y}_\ell)\boldsymbol{x} - \boldsymbol{b}\|_2$

$\boldsymbol{r}_\ell = \boldsymbol{A}(\boldsymbol{y}_\ell)\boldsymbol{x}_{\ell+1} - \boldsymbol{b}$

$\boldsymbol{D}^* = \boldsymbol{D}(\boldsymbol{x}_{\ell+1})$

compute the Jacobian $\boldsymbol{J}_\ell$ by (1.27)

$\boldsymbol{d}_\ell = \underset{\boldsymbol{d}}{\operatorname{argmin}} \|\boldsymbol{J}_\ell\boldsymbol{d} + \boldsymbol{r}_\ell\|_2$

determine steplength τ_ℓ

$\boldsymbol{y}_{\ell+1} = \boldsymbol{y}_\ell + \tau_\ell\boldsymbol{d}_\ell$

end

1.4 Alternating Optimization

In 1987 Bezdek et al. proposed the so-called *Grouped Coordinate Minimization* based on an alternating optimization of two subproblems, one over $\boldsymbol{x}$ and the other over $\boldsymbol{y}$. This

scheme has been later renamed as *Alternating Optimization* in [10], where the problem has been stated in a more general form, i.e. the objective function is a real valued function subject to nonlinear constraints and the variables have been partitioned in more than two subsets. In this thesis our discussion is restricted to two grouped subsets, both because this case can be easily extended to the more general and because it is of more interest for our applications. The method defines an iterative sequence $\{(\boldsymbol{x}_{\ell+1}, \boldsymbol{y}_{\ell+1})\} \in \mathbb{R}^n \times \mathbb{R}^m$ where

$$\begin{aligned} \boldsymbol{x}_{\ell+1} &= \underset{\boldsymbol{x}}{\operatorname{argmin}}\, f(\boldsymbol{x}, \boldsymbol{y}_\ell) \\ \boldsymbol{y}_{\ell+1} &= \underset{\boldsymbol{y}}{\operatorname{argmin}}\, f(\boldsymbol{x}_{\ell+1}, \boldsymbol{y}) \end{aligned} \tag{1.28}$$

that, under certain conditions, convergences to a local minimizer $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ of the objective function.

Computing an exact minimum point of $f(\cdot, \boldsymbol{x}_{\ell+1})$ and $f(\boldsymbol{y}_{\ell+1}, \cdot)$ is sometimes impractical, for instance for large scale problems. A typical way to overcome this issue consists of computing an approximation of those minimum points by applying an iterative method to both subproblems, stopping the iterations when some criterion is satisfied. For this reason, later formulations of the alternating approach investigate the accuracy required to solve the two subproblems in (1.28): in [33] one of the subproblems, let's say the one over $\boldsymbol{y}$, has been computed through one iteration of Newton's method; in [11] both $\boldsymbol{x}_{\ell+1}$ and $\boldsymbol{y}_{\ell+1}$ are obtained as inexact least squares solutions.

As a special case of the Alternating Optimization described in [9], we consider the least squares problem

$$\min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y}) = \|F(\boldsymbol{x}, \boldsymbol{y})\|_2^2 = \|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x} - \boldsymbol{b}\|_2^2 \tag{1.29}$$

where $\boldsymbol{b} \in \mathbb{R}^q$ is a measured noisy data vector, $\boldsymbol{y} \in \mathbb{R}^m$ is a vector of *nonlinear* parameters, $\boldsymbol{x} \in \mathbb{R}^n$ is a vector of *linear* variables and $\boldsymbol{A}$ is a nonlinear operator that maps $\boldsymbol{y}$ into a $q \times n$ matrix. Thus the iterative alternating scheme (1.28) becomes

$$\begin{aligned} \boldsymbol{x}_{\ell+1} &= \underset{\boldsymbol{x}}{\operatorname{argmin}}\, \|\boldsymbol{A}(\boldsymbol{y}_\ell)\boldsymbol{x} - \boldsymbol{b}\|_2 \\ \boldsymbol{y}_{\ell+1} &= \underset{\boldsymbol{y}}{\operatorname{argmin}}\, \|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x}_{\ell+1} - \boldsymbol{b}\|_2 \end{aligned} \tag{1.30}$$

In [9] Bezdek and Hathaway showed results about the convergence of the alternating scheme to the solution of the original problem. In particular, they pointed out that:

- by defining the sets of points that solve the least squares problem as

$$S_0 = \{(\boldsymbol{x}^*, \boldsymbol{y}^*) \in \mathbb{R}^n \times \mathbb{R}^m \text{ solutions of } (1.29)\}$$

and the limit points of the alternating scheme as

$$S = \{(\boldsymbol{x}^*, \boldsymbol{y}^*) \in \mathbb{R}^n \times \mathbb{R}^m \text{ solutions of (1.30)}\}$$

then $S_0 \subseteq S$. That is, the Alternating Optimization method can converge to a point that is not a solution of the original problem;

- in order to ensure global convergence of the Alternating Optimization, the iterates $\boldsymbol{x}_{\ell+1}$ and $\boldsymbol{y}_{\ell+1}$ in (1.30) must be in compact sets.

In the following we rewrite the results obtained in [9] for the least squares problem (1.29). The entire discussion about global convergence of the Alternating Optimization requires a crucial assumption, that is the existence and uniqueness of global minimizers for each of the subproblems. More precisely,

Assumption 1 (Existence and Uniqueness)**.** Let $f : \mathbb{R}^{n+m} \longrightarrow \mathbb{R}$ and let $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^n \times \mathbb{R}^m$. We assume that

$$h_1(\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{y})$$
$$h_2(\boldsymbol{y}) = f(\boldsymbol{x}, \boldsymbol{y})$$

have a unique global minimum.

Each subproblem defined by the alternating scheme is usually solved by an iterative method. Those algorithms are commonly based on the equations related to the first order necessary conditions, thus they seek stationary points of the objective function. By defining

$$E_0 = \{\boldsymbol{z}^* \in \mathbb{R}^{n+m} \text{ such that } \nabla f_{\boldsymbol{z}}(\boldsymbol{z}^*) = 0\}$$

and

$$E = \{\boldsymbol{z}^* = (\boldsymbol{x}^*, \boldsymbol{y}^*) \in \mathbb{R}^{n+m} \text{ such that } \nabla f_{\boldsymbol{x}}(\boldsymbol{z}^*) = 0, \nabla f_{\boldsymbol{y}}(\boldsymbol{z}^*) = 0\}$$

one can see that both the sets contain points for which $\nabla_{\boldsymbol{z}} f = \begin{bmatrix} \nabla_{\boldsymbol{x}} f \\ \nabla_{\boldsymbol{y}} f \end{bmatrix} = 0$. It follows that $E_0 = E$, that is the stationary points for the original problem (1.29) are the same as those for the subproblems in (1.30). Moreover, the set of points to which the Alternating Optimization converges, that is the set of stationary points for (1.30), contains vectors that behave like a global minimum when looking along one of the grouped subset of variables. Of course these vectors may not be global minimizers for f.
Before stating the following theorem, which describes the conditions necessary to ensure global convergence of the Alternating Optimization method and the set of its solutions, let us introduce the *iteration function* $\mathcal{T} : \mathbb{R}^{n+m} \to \mathbb{R}^{n+m}$, which generates a sequence $\{(\boldsymbol{x}_{\ell+1}, \boldsymbol{y}_{\ell+1}) = \mathcal{T}(\boldsymbol{x}_\ell, \boldsymbol{y}_\ell)\}$ of approximations of $(\boldsymbol{x}^*, \boldsymbol{y}^*)$.

Theorem 2. Let assumption 1 hold and let $(\boldsymbol{x}, \boldsymbol{y}) \in K_1 \times K_2$, where K_1, K_2 are compact sets in $\mathbb{R}^n$ and $\mathbb{R}^m$ respectively. By assuming $(\boldsymbol{x}_0, \boldsymbol{y}_0) \in K_1 \times K_2$ and by denoting with $M = \{\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^{n+m}$ such that $(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{T}(\boldsymbol{x}, \boldsymbol{y})\}$, then

- if $\boldsymbol{z} \in M$ then $\boldsymbol{z}^* = (\boldsymbol{x}^*, \boldsymbol{y}^*)$ with

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{A}(\boldsymbol{y}^*)\boldsymbol{x} - \boldsymbol{b}\|_2$$
$$\boldsymbol{y}^* = \underset{\boldsymbol{y}}{\operatorname{argmin}} \|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x}^* - \boldsymbol{b}\|_2$$

- $f(\boldsymbol{x}_{\ell+1}, \boldsymbol{y}_{\ell+1}) < f(\boldsymbol{x}_\ell, \boldsymbol{y}_\ell)$ and $f(\boldsymbol{x}_{\ell+1}, \boldsymbol{y}_{\ell+1}) = f(\boldsymbol{x}_\ell, \boldsymbol{y}_\ell)$ if and only if $(\boldsymbol{x}_\ell, \boldsymbol{y}_\ell) \in M$

- either

  - exists $(\boldsymbol{x}^*, \boldsymbol{y}^*) \in M$ and exists $\ell_0 \in \mathbb{N}$ such that $(\boldsymbol{x}_{\ell+1}, \boldsymbol{y}_{\ell+1}) = (\boldsymbol{x}^*, \boldsymbol{y}^*)$ for all $\ell > \ell_0$

  or

  - the limit of every convergent subsequence of $\{(\boldsymbol{x}_{\ell+1}, \boldsymbol{y}_{\ell+1})\}$ is in M.

1.4.1 Constrained Alternating Optimization

As we will see later, our interest concerns simple constraints imposed on one or both the subsets of variables. The Alternating Optimization method is affected by the constraints just for the iterative method used to solve the subproblems, which must take into account the constrained formulation. The original problem will be

$$\min_{\boldsymbol{z} \in \Omega} f(\boldsymbol{z})$$

where $\Omega \subseteq \mathbb{R}^p$ is the feasible region, which for simple constraints can be written as

$$\Omega = \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^n \times \mathbb{R}^m \text{ such that } l_{\boldsymbol{x}} \leq \boldsymbol{x} \leq u_{\boldsymbol{x}}, l_{\boldsymbol{y}} \leq \boldsymbol{y} \leq u_{\boldsymbol{y}}\}$$

By making the separability explicit, the original problem becomes

$$\min_{\boldsymbol{x} \in \Omega_1, \boldsymbol{y} \in \Omega_2} \|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x} - \boldsymbol{b}\|_2^2,$$

where $\Omega_1 = \{\boldsymbol{x} : l_{\boldsymbol{x}} \leq \boldsymbol{x} \leq u_{\boldsymbol{x}}\}$, $\Omega_2 = \{\boldsymbol{y} : l_{\boldsymbol{y}} \leq \boldsymbol{y} \leq u_{\boldsymbol{y}}\}$, while the alternating scheme is

Alternating Optimization to solve: $\min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{x},\boldsymbol{y})$

choose initial $\boldsymbol{y}_0$

for $\ell = 0, 1, 2, \ldots$

$$\boldsymbol{x}_{\ell+1} = \operatorname*{argmin}_{\boldsymbol{x}\in\Omega_1} \|\boldsymbol{A}(\boldsymbol{y}_\ell)\boldsymbol{x} - \boldsymbol{b}\|_2$$
$$\boldsymbol{y}_{\ell+1} = \operatorname*{argmin}_{\boldsymbol{y}\in\Omega_2} \|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x}_{\ell+1} - \boldsymbol{b}\|_2$$

end

Theorem 2 can be rewritten as

Theorem 3. Let assumption 1 hold and let $(\boldsymbol{x},\boldsymbol{y}) \in \Omega_1 \times \Omega_2$, compact sets in $\mathbb{R}^n$ and $\mathbb{R}^m$ respectively. By assuming $(\boldsymbol{x}_0,\boldsymbol{y}_0) \in \Omega_1 \times \Omega_2$ and by denoting with $M = \{\boldsymbol{z} = (\boldsymbol{x},\boldsymbol{y}) \in \Omega_1 \times \Omega_2$ such that $(\boldsymbol{x},\boldsymbol{y}) = \mathcal{T}(\boldsymbol{x},\boldsymbol{y})\}$, then

- if $\boldsymbol{z} \in M$ then $\boldsymbol{z}^* = (\boldsymbol{x}^*,\boldsymbol{y}^*)$ with

$$\boldsymbol{x}^* = \operatorname*{argmin}_{\boldsymbol{x}\in\Omega_1} \|\boldsymbol{A}(\boldsymbol{y}^*)\boldsymbol{x} - \boldsymbol{b}\|_2$$
$$\boldsymbol{y}^* = \operatorname*{argmin}_{\boldsymbol{y}\in\Omega_2} \|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x}^* - \boldsymbol{b}\|_2$$

- $f(\boldsymbol{x}_{\ell+1},\boldsymbol{y}_{\ell+1}) < f(\boldsymbol{x}_\ell,\boldsymbol{y}_\ell)$ and $f(\boldsymbol{x}_{\ell+1},\boldsymbol{y}_{\ell+1}) = f(\boldsymbol{x}_\ell,\boldsymbol{y}_\ell)$ if and only if $(\boldsymbol{x}_\ell,\boldsymbol{y}_\ell) \in M$

- either

  - exists $(\boldsymbol{x}^*,\boldsymbol{y}^*) \in M$ and exists $\ell_0 \in \mathbb{N}$ such that $(\boldsymbol{x}_{\ell+1},\boldsymbol{y}_{\ell+1}) = (\boldsymbol{x}^*,\boldsymbol{y}^*)$ for all $\ell > \ell_0$

  or

  - the limit of every convergent subsequence of $\{(\boldsymbol{x}_{\ell+1},\boldsymbol{y}_{\ell+1})\}$ is in M.

We recall that, as in the unconstrained case, the points to which Alternating Optimization converges can be global minimizers of f, but also local minimizers or saddle points.

1.5 Comparison between the two approaches

In this section we compare the Variable Projection and the Alternating Optimization methods, pointing out common features and main differences. The key idea on which both methods are based is to exploit the separability of the initial variable $\boldsymbol{z} \in \mathbb{R}^p$ in order to simplify the original problem: the initial joint optimization is replaced by reduced easier subproblems. In particular, we focus on the special case when one of the subsets of variables is linear and subject to nonnegative constraints. The advantage given by the linear dependence of $F(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{A}(\boldsymbol{y})\boldsymbol{x} - \boldsymbol{b}$ over $\boldsymbol{x}$ is to obtain, for both methods, a linear least squares subproblem, which is obviously preferable both from a numerical point of view and because its solution can be written through a closed formula.

The first difference between the two approaches is that Variable Projection, through an explicit elimination of the linear variable $\boldsymbol{x}$, modifies the objective function, which becomes dependent only on the other subset of variables $\boldsymbol{y}$:

$$\min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y}) \qquad \longmapsto \qquad \min_{\boldsymbol{y}} \psi(\boldsymbol{y}) = f(\boldsymbol{x}(\boldsymbol{y}), \boldsymbol{y})$$

This makes the main least squares problem, in Variable Projection, the one over the $\boldsymbol{y}$ variable, thus a nonlinear least squares. An inner linear least squares still occurs though, implicitly due to the residual evaluation (as seen in (1.25) in section 1.3.2). On the other hand, in Alternating Optimization one splits the minimization of $f(\boldsymbol{x}, \boldsymbol{y})$ over the two separated variables, $\boldsymbol{x}$ and $\boldsymbol{y}$, thus leading to a linear and a nonlinear least squares problem.

We gather again the algorithms of the two methods:

Variable Projection to solve: $\min_{\boldsymbol{y}} \psi(\boldsymbol{y}) = f(\boldsymbol{x}(\boldsymbol{y}), \boldsymbol{y})$

choose initial $\boldsymbol{y}_0$

for $\ell = 0, 1, 2, \ldots$

$\boldsymbol{x}_{\ell+1} = \underset{\boldsymbol{x} \geq \boldsymbol{0}}{\operatorname{argmin}} \|\boldsymbol{A}(\boldsymbol{y}_\ell)\boldsymbol{x} - \boldsymbol{b}\|_2$

$\boldsymbol{r}_\ell = \boldsymbol{A}(\boldsymbol{y}_\ell)\boldsymbol{x}_{\ell+1} - \boldsymbol{b}$

$\boldsymbol{D}^* = \boldsymbol{D}(\boldsymbol{x}_{\ell+1})$

compute the Jacobian $\boldsymbol{J}_\ell$ by (1.27)

$\boldsymbol{d}_\ell = \underset{\boldsymbol{d}}{\operatorname{argmin}} \|\boldsymbol{J}_\ell \boldsymbol{d} + \boldsymbol{r}_\ell\|_2$

determine steplength τ_ℓ

$\boldsymbol{y}_{\ell+1} = \boldsymbol{y}_\ell + \tau_\ell \boldsymbol{d}_\ell$

end

Alternating Optimization to solve: $\min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y})$

choose initial $\boldsymbol{y}_0$

for $\ell = 0, 1, 2, \ldots$

$\boldsymbol{x}_{\ell+1} = \underset{\boldsymbol{x} \geq 0}{\operatorname{argmin}} \|\boldsymbol{A}(\boldsymbol{y}_\ell)\boldsymbol{x} - \boldsymbol{b}\|_2$

$\boldsymbol{y}_{\ell+1} = \underset{\boldsymbol{y}}{\operatorname{argmin}} \|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x}_{\ell+1} - \boldsymbol{b}\|_2$

end

The main difference between Variable Projection and Alternating Optimization lays on the related objective functions and, consequently, their derivatives. In fact least squares are commonly solved by iterative methods, based on necessary conditions and consequently on zeroing the gradient of the objective function. Thus:

Variable Projection: the evaluation of $\nabla_{\boldsymbol{y}}\psi$ requires an implicit differentiation of $\boldsymbol{x}(\boldsymbol{y}) = (\boldsymbol{A}(\boldsymbol{y})\boldsymbol{D}^*)^\dagger \boldsymbol{b}$;

Alternating Optimization: in order to solve the subproblems in (1.30) we need separately $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{y})$ and $\nabla_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y})$.

In chapters 3 and 4, $\boldsymbol{x}_{\ell+1}$ will be computed through projection methods; on the other hand, as regarding Alternating Optimization, $\boldsymbol{y}_{\ell+1}$ will be obtained through one step of Gauss-Newton algorithm. Especially in this case, the differences between the two approaches could be confusing; in order to explain them clearly, the following scheme contains each step of Variable Projection - on the left - and Alternating Optimization (with one step of Gauss-Newton for the $\boldsymbol{y}$ subproblem) - on the right.

Variable Projection	**Alternating Optimization**
given $\boldsymbol{y}_0$	given $\boldsymbol{y}_0$
for $\ell = 0, 1, 2, \ldots$	for $\ell = 0, 1, 2, \ldots$
1. $\boldsymbol{x}_{\ell+1} = \underset{\boldsymbol{x}\geq 0}{\operatorname{argmin}} \lVert \boldsymbol{A}(\boldsymbol{y}_\ell)\boldsymbol{x} - \boldsymbol{b} \rVert_2$	1. $\boldsymbol{x}_{\ell+1} = \underset{\boldsymbol{x}\geq 0}{\operatorname{argmin}} \lVert \boldsymbol{A}(\boldsymbol{y}_\ell)\boldsymbol{x} - \boldsymbol{b} \rVert_2$
2. $\boldsymbol{r}_\ell = \boldsymbol{A}(\boldsymbol{y}_\ell)\boldsymbol{x}_{\ell+1} - \boldsymbol{b}$	2. $\boldsymbol{r}_\ell = \boldsymbol{A}(\boldsymbol{y}_\ell)\boldsymbol{x}_{\ell+1} - \boldsymbol{b}$
3. $\boldsymbol{J}_{VP} = \boldsymbol{J}_{\boldsymbol{y}}(\psi(\boldsymbol{y})) = \boldsymbol{J}(f(\boldsymbol{x}(\boldsymbol{y}), \boldsymbol{y}))$	3. $\boldsymbol{J}_{AO} = \boldsymbol{J}_{\boldsymbol{y}}(f(\boldsymbol{x}, \boldsymbol{y}))$
4. $\boldsymbol{d}_\ell = \underset{\boldsymbol{d}}{\operatorname{argmin}} \lVert \boldsymbol{J}_{VP}\boldsymbol{d} + \boldsymbol{r}_\ell \rVert_2$	4. $\boldsymbol{d}_\ell = \underset{\boldsymbol{d}}{\operatorname{argmin}} \lVert \boldsymbol{J}_{AO}\boldsymbol{d} + \boldsymbol{r}_\ell \rVert_2$
5. determine τ_ℓ	5. determine τ_ℓ
6. $\boldsymbol{y}_{\ell+1} = \boldsymbol{y}_\ell + \tau_\ell \boldsymbol{d}_\ell$	6. $\boldsymbol{y}_{\ell+1} = \boldsymbol{y}_\ell + \tau_\ell \boldsymbol{d}_\ell$
end	end

It's clear that the crucial point is in the Jacobian matrix. For the Alternating Optimization method, given the residual $\boldsymbol{r}(\boldsymbol{y}) = \boldsymbol{A}(\boldsymbol{y})\boldsymbol{x} - \boldsymbol{b}$, the k-th column of the Jacobian $\boldsymbol{J}_{AO}$ is

$$(\boldsymbol{J}_{AO})_k = \frac{\partial \boldsymbol{r}(\boldsymbol{x}, \boldsymbol{y})}{\partial y_k} = \frac{\partial \boldsymbol{A}(\boldsymbol{y})\boldsymbol{x}}{\partial y_k}. \tag{1.31}$$

For the Variable Projection, as already seen, the computation of the Jacobian is more complicated. Indeed, due to the implicit dependence of $\boldsymbol{x}$ over $\boldsymbol{y}$, the product rule makes the differentiation of the residual $\boldsymbol{r}(\boldsymbol{y}) = \boldsymbol{A}(\boldsymbol{y})\left(\boldsymbol{A}(\boldsymbol{y})\boldsymbol{D}^*\right)^\dagger \boldsymbol{b} - \boldsymbol{b}$ composed by two terms.

The k-th column of $\boldsymbol{J}_{VP}$ is

$$(\boldsymbol{J}_{VP})_k = \frac{\partial \boldsymbol{r}(\boldsymbol{y})}{\partial y_k} = \frac{\partial}{\partial y_k}\left(\boldsymbol{A}(\boldsymbol{y})\left(\boldsymbol{A}(\boldsymbol{y})\boldsymbol{D}^*\right)^\dagger \boldsymbol{b}\right) = \tag{1.32}$$
$$= \frac{\partial \boldsymbol{A}(\boldsymbol{y})}{\partial y_k}\left(\boldsymbol{A}(\boldsymbol{y})\boldsymbol{D}^*\right)^\dagger \boldsymbol{b} + \boldsymbol{A}(\boldsymbol{y})\frac{\partial\left(\boldsymbol{A}(\boldsymbol{y})\boldsymbol{D}^*\right)^\dagger \boldsymbol{b}}{\partial y_k}\,.$$

The second term involves the differentiation of the pseudoinverse matrix, while the first is in a certain sense comparable to $\boldsymbol{J}_{AO}$.

A further difference is that the linear solution $\boldsymbol{x}_{\ell+1}$ in step 1 can be inexact for the Alternating Optimization, while the computation of $\boldsymbol{D}^*$ in Variable Projection requires the solution of the linear least squares to be exact.

We summarize now the differences between Variable Projection and Alternating Optimization:

Objective function: Variable Projection modifies the objective function, thus there is a sensitive difference in the computation of the derivatives with Alternating Optimization.

Accuracy of $\boldsymbol{x}$: $\boldsymbol{x}_{\ell+1}$ in step 1 may be inexact in Alternating Optimization method, while in Variable Projection, in order to compute $\boldsymbol{D}^*$, must be the exact solution of $\min_{\boldsymbol{x}} \|\boldsymbol{A}(\boldsymbol{y}_\ell)\boldsymbol{x} - \boldsymbol{b}\|_2^2$.

$\boldsymbol{y}$ dimension: When $\boldsymbol{y}$ is relatively large the computation of $\boldsymbol{J}_{VP}$ could be really expensive. In this case the Alternating Optimization method may be preferable. On the other hand, when $\boldsymbol{y}$ is small, the Variable Projection approach performs computationally better.

1.5.1 A numerical experiment

We have implemented a simple Gaussian fitting example, taken from [44] and modified by adding nonnegative constraints on $\boldsymbol{x}$. Given $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^m$ and t is a n-vector of equispaced sample points between 0 and 1, let us assume that the model function describing the data $\boldsymbol{b} \in \mathbb{R}^q$ is

$$\eta(\boldsymbol{x}, \boldsymbol{y}; t) = x_1 \exp(-y_1 t) + x_2 \exp(-y_2(t - y_5)^2) +$$
$$+ x_3 \exp(-y_3(t - y_6)^2) + x_4 \exp(-y_4(t - y_7)^2)\,,$$

where $n = 4$, $m = 7$ and $q = 64$. Thus the residual is $\boldsymbol{r}(\boldsymbol{x}, \boldsymbol{y}) = \eta(\boldsymbol{x}, \boldsymbol{y}; t) - \boldsymbol{b}$ or, equivalently, $\boldsymbol{r}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{A}(\boldsymbol{y})\boldsymbol{x} - \boldsymbol{b}$, where $\boldsymbol{A}$ is the $q \times n$ matrix defined by

$$\boldsymbol{A}(\boldsymbol{y}) = \begin{bmatrix} \exp(-y_1 t_1) & \exp(-y_2(t_1 - y_5)^2) & \exp(-y_3(t_1 - y_6)^2) & \exp(-y_4(t_1 - y_7)^2) \\ \vdots & \vdots & \vdots & \vdots \\ \exp(-y_1 t_q) & \exp(-y_2(t_q - y_5)^2) & \exp(-y_3(t_q - y_6)^2) & \exp(-y_4(t_q - y_7)^2) \end{bmatrix}.$$

The Gaussian fitting example can be thus written as a separable problem of the form

$$\min_{\boldsymbol{x},\boldsymbol{y}} \|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x} - \boldsymbol{b}\|_2^2 \text{ subject to } \boldsymbol{x} \geq 0$$

that in the following we solve by applying the Variable Projection and the Alternating Optimization.

We know from (1.31) and (1.32) that to compute the Jacobian $\boldsymbol{J}_{AO}$ we need $\dfrac{\partial \boldsymbol{A}}{\partial y_k}$; while for $\boldsymbol{J}_{VP}$ also $\dfrac{\partial \left(\boldsymbol{A}\boldsymbol{D}^*\right)^\dagger}{\partial y_k}$ is required. Let us proceed by step. For a given index k, $\dfrac{\partial \boldsymbol{A}}{\partial y_k}$ is a $q \times n$ matrix, whose i-th column contains all zeros where $(\boldsymbol{A})_i$ (the i-th column of $\boldsymbol{A}$) does not depend on y_k. Thus:

$$\frac{\partial \boldsymbol{A}}{\partial y_1} = \begin{bmatrix} -t_1 \exp(-y_1 t_1) & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ -t_q \exp(-y_1 t_q) & 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial \boldsymbol{A}}{\partial y_2} = \begin{bmatrix} 0 & -(t_1 - y_5)^2 \exp\left(-y_2(t_1 - y_5)^2\right) & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & -(t_q - y_5)^2 \exp\left(-y_2(t_q - y_5)^2\right) & 0 & 0 \end{bmatrix}$$

$$\frac{\partial \boldsymbol{A}}{\partial y_3} = \begin{bmatrix} 0 & 0 & -(t_1 - y_6)^2 \exp\left(-y_3(t_1 - y_6)^2\right) & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & -(t_q - y_6)^2 \exp\left(-y_3(t_q - y_6)^2\right) & 0 \end{bmatrix}$$

$$\frac{\partial \boldsymbol{A}}{\partial y_4} = \begin{bmatrix} 0 & 0 & 0 & -(t_1 - y_7)^2 \exp\left(-y_4(t_1 - y_7)^2\right) \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -(t_q - y_7)^2 \exp\left(-y_4(t_q - y_7)^2\right) \end{bmatrix}$$

$$\frac{\partial \boldsymbol{A}}{\partial y_5} = \begin{bmatrix} 0 & 2y_2(t_1 - y_5) \exp\left(-y_2(t_1 - y_5)^2\right) & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 2y_2(t_q - y_5) \exp\left(-y_2(t_q - y_5)^2\right) & 0 & 0 \end{bmatrix}$$

$$\frac{\partial \boldsymbol{A}}{\partial y_6} = \begin{bmatrix} 0 & 0 & 2y_3(t_1 - y_6)\exp\left(-y_3(t_1 - y_6)^2\right) & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 2y_3(t_q - y_6)\exp\left(-y_3(t_q - y_6)^2\right) & 0 \end{bmatrix}$$

$$\frac{\partial \boldsymbol{A}}{\partial y_7} = \begin{bmatrix} 0 & 0 & 0 & 2y_4(t_1 - y_7)\exp\left(-y_4(t_1 - y_7)^2\right) \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 2y_4(t_q - y_7)\exp\left(-y_4(t_q - y_7)^2\right) \end{bmatrix}$$

It follows that the k-th column of $\boldsymbol{J}_{AO}$ is the q-dimensional vector defined by

$$(\boldsymbol{J}_{AO})_k = \frac{\partial \boldsymbol{A}}{\partial y_k}\boldsymbol{x}.$$

As regards $\boldsymbol{J}_{VP}$, equation (1.32) shows that it is composed by two terms: the first involving $\dfrac{\partial \boldsymbol{A}}{\partial y_k}$, the second $\dfrac{\partial \left(\boldsymbol{A}\boldsymbol{D}^*\right)^\dagger}{\partial y_k}$. Here we don't provide an explicit expression of $\boldsymbol{J}_{VP}$, but just remind that the differentiation of a pseudoinverse $\boldsymbol{M}^\dagger$ is

$$\frac{\partial \boldsymbol{M}^\dagger}{\partial \boldsymbol{y}} = (\boldsymbol{M}^T\boldsymbol{M})^{-1}\frac{\partial \boldsymbol{M}^T}{\partial \boldsymbol{y}}\left(\boldsymbol{I} - \boldsymbol{M}\boldsymbol{M}^\dagger\right) - \boldsymbol{M}^\dagger\frac{\partial \boldsymbol{M}}{\partial \boldsymbol{y}}\boldsymbol{M}^\dagger\,,$$

and we recall equation (1.27). In conclusion, $\boldsymbol{J}_{VP}$ can be computed by means of $(\boldsymbol{A}(\boldsymbol{y})\boldsymbol{D}^*)^\dagger$, $\dfrac{\partial \boldsymbol{A}}{\partial y_k}$ and its transpose.

Here we create the test problem as follows. Set $\boldsymbol{x}_{true} = (5, 18, 15, 10)^T$ and $\boldsymbol{y}_{true} = \left(10, \dfrac{1}{0.015}, \dfrac{1}{0.03}, \dfrac{1}{0.015}, 0.25, 0.5, 0.75\right)^T$, the measured data are constructed as

$$\boldsymbol{b} = \boldsymbol{A}(\boldsymbol{y}_{true})\boldsymbol{x}_{true} + \boldsymbol{\eta}\,,$$

where $\boldsymbol{\eta}$ is normally distributed random noise with standard deviation equal to 1.

We used the projected Landweber method to solve the linear problem and the Levenberg-Marquardt to compute the nonlinear solution. For implementation details on these methods we refer to [38]. We stopped the iterative schemes when a maximum number of cycles $\ell = 10$ has been performed. The reconstructed fittings in Figure 1.1 show that Variable Projection seems to provide a better solution. In Figure 1.2 the relative errors on $\boldsymbol{x}$ and $\boldsymbol{y}$ at each iteration ℓ are given.

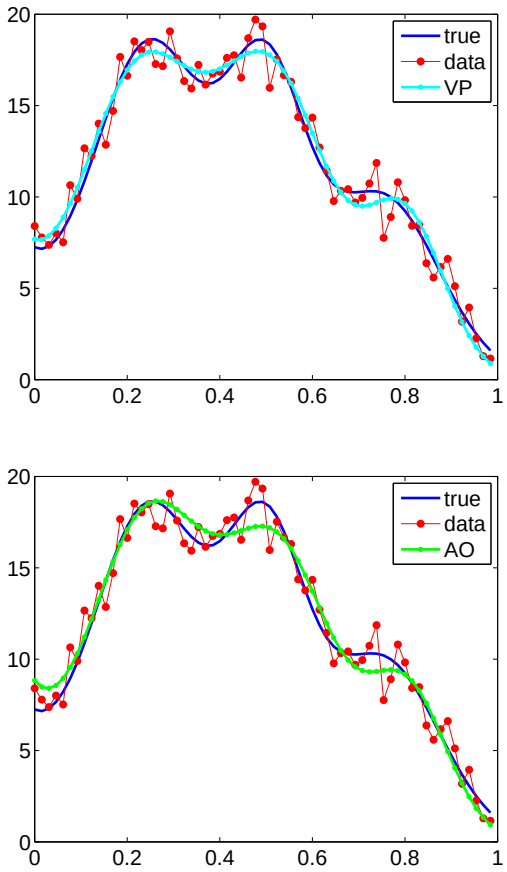


Figure 1.1: Variable Projection (on the top) and Alternating Optimization (on the bottom) reconstructions.

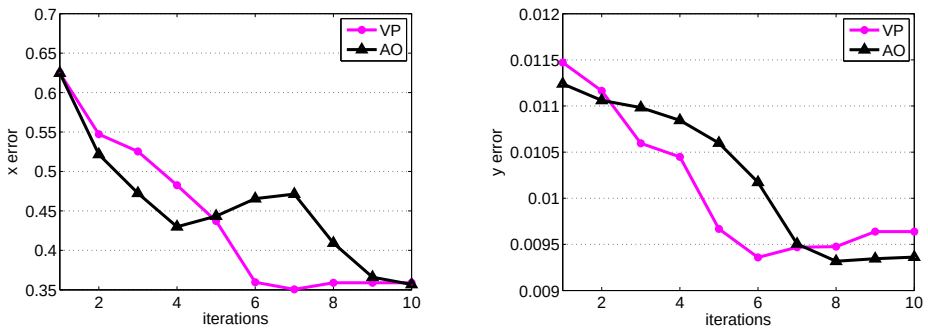


Figure 1.2: Relative errors on $\boldsymbol{x}$ (on the left) and $\boldsymbol{y}$ (on the right) obtained with Variable Projection (magenta dots) and Alternating Optimization (black triangles).

Chapter 2

Image deconvolution

The image formation process derives from the discretization of a Fredholm integral equation of the first kind, thus is severely affected by ill-posedness. Furthermore, the presence of noise in the measured data makes any naive inversion strategy useless, and only an accurate study of the instability affecting each problem together with a choice of an appropriate regularization technique to address its solution can lead to a physically meaningful result.

This chapter is organized as follows: in section 2.1 we describe the image formation process and how to overcome the deriving ill-posedness; later in section 2.1 we discuss how imposing nonnegative constraints can improve the reconstruction; then we propose a new Regularized Scaled Gradient Projection algorithm in section 2.3 and finally we give some numerical results in section 2.4.

2.1 Ill-posedness and regularization

An image can be thought as a degraded signal of the original object, not directly observable [6]. During its acquisition and registration, the signal is corrupted by two kinds of degradation. One is related to the instrumental setting and is referred to as blurring; for example, in the case of an astronomical image, it is due to atmospheric turbulence or related motion between camera and ground. The second one, the so-called noise, occurs during the recording process and is due, for instance, to the analog to digital conversion of the signal. Therefore the measured data image $\boldsymbol{b} \in \mathbb{R}^n$ is obtained by adding the noise contribution $\boldsymbol{\eta} \in \mathbb{R}^n$ to the convolution between the point spread function (PSF) $\boldsymbol{\mathcal{K}}$, describing the blurring effect, and the true object $\boldsymbol{x} \in \mathbb{R}^n$. One simple and common

mathematical model to describe the image formation process is the convolution equation

$$\boldsymbol{b} = \boldsymbol{\mathcal{K}} * \boldsymbol{x} + \boldsymbol{\eta}\,, \tag{2.1}$$

which assumes that the blur is spatially invariant. For instance, the PSF for the blurring caused by atmospheric turbulence can be described as the two-dimensional Gaussian function. The convolution operator associated with the PSF can be discretized through a structured matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ so that equation (2.1) in discrete form becomes [6, 29, 30, 31]

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\eta}\,.$$

The problem of image reconstruction consists of recovering the unknown object $\boldsymbol{x}$ when the measured data $\boldsymbol{b}$ is given and can be formulated as the solution of the equation

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}\,. \tag{2.2}$$

Sometimes the matrix $\boldsymbol{A}$ associated to the PSF can be estimated thus assumed to be known: in that case equation (2.2) is referred to as *deconvolution* problem. But when both the object and the PSF are unknown while the only data $\boldsymbol{b}$ is given, we talk about *blind deconvolution*.

In both cases, (2.2) is a large scale problem thus it is numerically impossible to compute the solution by inverting the matrix $\boldsymbol{A}$. Moreover, even if we assume $\boldsymbol{A}$ nonsingular, since $\boldsymbol{A}$ is typically severely ill-conditioned, the naive solution $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$ wouldn't be meaningful due to the presence of noise on the measured data. A typical way to overcome this situation consists of considering the least squares problem

$$\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \tag{2.3}$$

and exploiting appropriate regularization techniques to address its solution in a numerically stable way.

A classical approach to regularization, the so-called *direct regularization* [29], is to incorporate further a priori information about the expected solution. An example is the well-known Tikhonov regularization scheme, which consists of adding, to the fit-to-data term in (2.3), a smoothing assumption on the solution, thus leading to the optimization problem

$$\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda^2\|\boldsymbol{x}\|_2^2\,. \tag{2.4}$$

The role of the regularization parameter λ is to balance the solution between the discrepancy to the data and the smoothing request. Several rules to estimate λ have been proposed in literature, such as the L-curve principle [32] and Generalized Cross Validation (GCV) [29, 30, 31, 55].

Another regularization approach, referred to as *iterative regularization* [29, 55], consists of iteratively building up a sequence of arrays that converges to the solution of (2.3), and stopping the iterative procedure before the noise introduces artifacts and undesired effects on the reconstructions. The choice of the iterative method to be used to solve (2.3) strongly depends of the matrix $\boldsymbol{A}$ and the instability level of the problem, and typically involves classical approaches like gradient algorithms or the Newton method.

A priori information on the unknown image can be easily handled in both these kinds of regularization. For example, it is natural in many applications, like astronomical imaging, to impose a nonnegativity constraint on the pixels content. In this case, problems (2.3) and (2.4) can be turned into the corresponding constrained minimization problems, which can be solved by descent methods including a projection step at each iteration.

2.2 Role of nonnegative constraints

In many imaging applications, nonnegative constraints are motivated by the fact that the entries of the vector $\boldsymbol{x}$ are the components of a grayscale digital image, which must be nonnegative. In this section we show how imposing nonnegative constraints to the deconvolution problem (2.3) significantly improve the quality of the reconstructed image [55]. We compare the unconstrained formulation (2.3), solved with CGLS [29, 31], with the nonnegative formulation

$$\min_{\boldsymbol{x}\geq 0} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \,,$$

solved by performing the Scaled Gradient Projection (SGP) method proposed in [5].

In the following we show reconstructions of the satellite image from *Restore Tools*[1], corrupted by additive Gaussian noise with zero mean and variance equal to 1 and scaled so that $\dfrac{||\boldsymbol{\eta}||}{||\boldsymbol{A}\boldsymbol{x}||} = 0.01$.
The results show that, despite the fact that the relative errors are similar (Figure 2.1), the constrained formulation produces an improved reconstructed image, especially in the background (Figure 2.2).

Moreover, we remark that in [4] the authors provide a theoretical explanation of the well-known stabilizing effect given by the incorporation of nonnegative constraint in image reconstruction. Indeed, reminding that the unconstrained problem is

$$\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$$

[1] http://www.mathcs.emory.edu/∼nagy/RestoreTools/

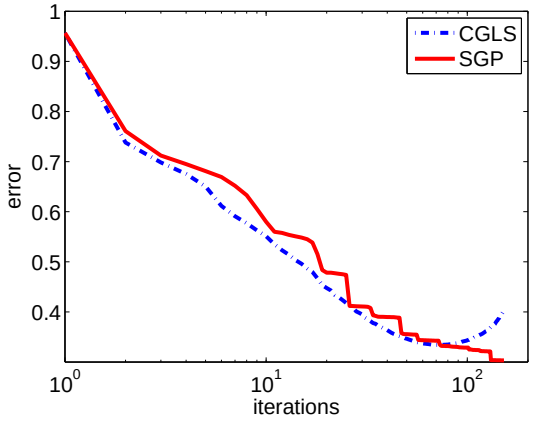


Figure 2.1: Comparison between unconstrained and nonnegative formulations: relative errors of CGLS and SGP.

while the nonnegative least squares can be formulated as

$$\min_{\boldsymbol{x}\geq\boldsymbol{0}} \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2 = \min_{\boldsymbol{x}} \|(\boldsymbol{A}\boldsymbol{D}^*)\boldsymbol{x}-\boldsymbol{b}\|_2^2,$$

with $\boldsymbol{D}^*$ defined in (1.22), they prove that the condition number of $\boldsymbol{A}$ is higher than the one of $\boldsymbol{A}\boldsymbol{D}^*$.

2.3 Projection methods for nonnegative linear least squares

In this section we present a scaled gradient projection method suitable for large scale linear ill-posed problems, such as image deconvolution. In particular, our interest is to address a nonnegative linear least squares formulation, such as

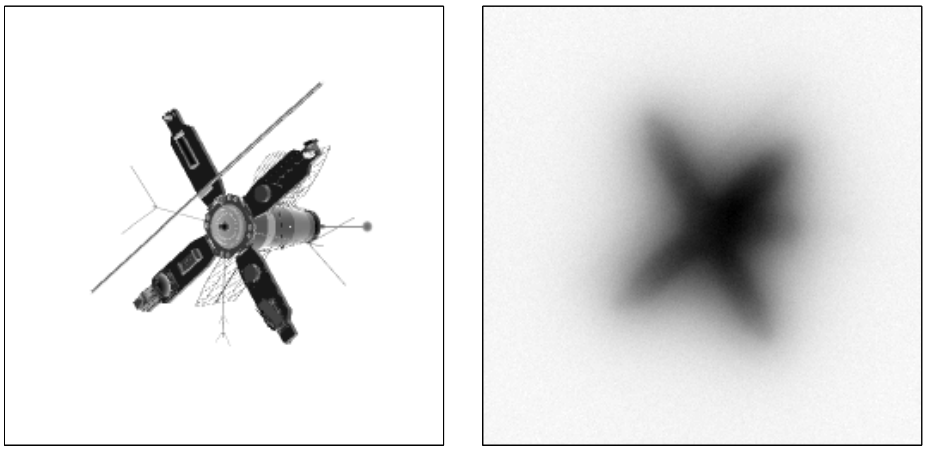
$$\min_{\boldsymbol{x}\geq\boldsymbol{0}} h(\boldsymbol{x}) = \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2\,, \tag{2.5}$$

where $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ is nonsingular and $\boldsymbol{x},\boldsymbol{b}\in\mathbb{R}^n$. Given a current iterate $\boldsymbol{x}_k$, the Scaled Gradient-Projection algorithm [8, 38, 55] is described by

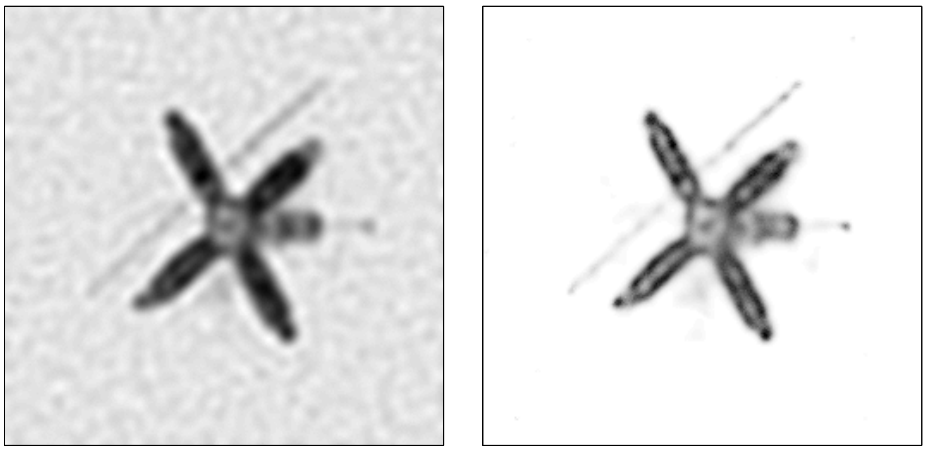
$$\boldsymbol{x}_{k+1} = \mathcal{P}\left(\boldsymbol{x}_k - \tau_k \boldsymbol{M}_k \boldsymbol{g}_k\right)\,,$$

where:

- $\boldsymbol{g}_k = \nabla h(\boldsymbol{x}_k) = 2\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x}_k-\boldsymbol{b})$ is the gradient direction evaluated at the current iterate;



Exact image Blurred noisy image



CGLS SGP

Figure 2.2: First row: from the left, the exact image to recover and the measured data. Second row: CGLS solution of the unconstrained formulation and nonnegative reconstruction through SGP at the iteration corresponding to the minimum error.

- τ_k is the steplength parameter that must ensure sufficient decrease of the objective function h;
- $\boldsymbol{M}_k$ is a positive definite matrix scaling the gradient;
- $\mathcal{P}$ is the projection operator, here defined by $\mathcal{P}(x_i) = \max(0, x_i)\,, i = 1, \ldots, n$.

There is a large variety of possibilities for choosing the scaling matrix $\boldsymbol{M}_k$, whose main contribution is to accelerate the slow convergence rate provided by the projected gradient direction, by approximating in some way the inverse of the Hessian matrix $\boldsymbol{H} = 2\boldsymbol{A}^T\boldsymbol{A}$. For this reason, the Scaled Gradient-Projection algorithm is also referred to as Quasi Projected-Newton.
We point out that in order to obtain the Projected-Newton method, one cannot simply choose $\boldsymbol{M}_k = (2\boldsymbol{A}^T\boldsymbol{A})^{-1}$. Indeed the Hessian must be replaced by the so-called *reduced Hessian* $\boldsymbol{H}^R_k = (\boldsymbol{H}^R)_{ij}(\boldsymbol{x}_k)$, defined by

$$(\boldsymbol{H}^R)_{ij}(\boldsymbol{x}) = \begin{cases} \delta_{ij} & \text{if } i \in \mathcal{A}(\boldsymbol{x}) \quad \text{or} \quad \text{if } j \in \mathcal{A}(\boldsymbol{x}) \\ (2\boldsymbol{A}^T\boldsymbol{A})_{ij} & \text{otherwise} \end{cases}$$

or, equivalently, by $\boldsymbol{H}^R = 2\boldsymbol{D}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{D} + (\boldsymbol{I} - \boldsymbol{D})$, where $\boldsymbol{D}$ is the diagonal matrix defined in (1.22). Similarly, one can define the *reduced gradient* as $\boldsymbol{g}^R(\boldsymbol{x}) = \boldsymbol{D}\boldsymbol{g}(\boldsymbol{x}) = 2\boldsymbol{D}\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})$.

In general, the Projected-Newton algorithm can be written as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \tau_k \boldsymbol{p}_k\,,$$

where the descent direction is

$$\boldsymbol{p}_k = -(\boldsymbol{H}^R_k)^{-1}\boldsymbol{g}_k\,. \tag{2.6}$$

The previous expression suggests the possibility to choose the direction by restricting the computation to the r free variables and then extend the resulting vector by adding $n - r$ zeros corresponding to the non-free components. In the following we show that the resulting vector $\boldsymbol{p}^R_k$ is a descent direction for h in (2.5).

Let us assume for simplicity that $\boldsymbol{D} = \begin{bmatrix} \boldsymbol{I}_r & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$; furthermore, let us partition the Hessian into corresponding blocks, that is

$$\boldsymbol{H} = \begin{matrix} \\ r \\ n-r \end{matrix} \overset{\begin{matrix} r & n-r \end{matrix}}{\begin{bmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_2 \\ \boldsymbol{H}_3 & \boldsymbol{H}_4 \end{bmatrix}}$$

As a consequence, the reduced Hessian will be written as $\boldsymbol{H}^R = \begin{bmatrix} \boldsymbol{H}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{n-r} \end{bmatrix}$. At the same way as $\boldsymbol{H}$, the gradient vector can be partitioned as $\boldsymbol{g} = \begin{bmatrix} \boldsymbol{g}_1 \\ \boldsymbol{g}_2 \end{bmatrix}$, which allows to write the reduced gradient as $\boldsymbol{g}^R = \boldsymbol{D}\boldsymbol{g} = \begin{bmatrix} \boldsymbol{g}_1 \\ \boldsymbol{0} \end{bmatrix}$.

The proposed direction is

$$\boldsymbol{p}_k^R = -(\boldsymbol{H}^R)_k^{-1}\boldsymbol{g}_k^R = -\begin{bmatrix} (\boldsymbol{H}_1)_k^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{n-r} \end{bmatrix}\begin{bmatrix} (\boldsymbol{g}_1)_k \\ \boldsymbol{0} \end{bmatrix} = -\begin{bmatrix} (\boldsymbol{H}_1)_k^{-1}(\boldsymbol{g}_1)_k \\ \boldsymbol{0} \end{bmatrix}. \tag{2.7}$$

Due to the fact that any submatrix of a positive definite matrix obtained by deleting a certain number of rows and corresponding columns is still positive definite, the assumption of $\boldsymbol{A}$ positive definite ensures also $\boldsymbol{H}$ and $\boldsymbol{H}_1$ to be positive definite thus invertible. It follows from trivial computation that (2.7) is a descent direction for h; indeed at each iteration k it holds that

$$\nabla h(\boldsymbol{x}_k)^T\boldsymbol{p}_k^R = -\begin{bmatrix} (\boldsymbol{g}_1)_k & (\boldsymbol{g}_2)_k \end{bmatrix}\begin{bmatrix} (\boldsymbol{H}_1)_k^{-1}(\boldsymbol{g}_1)_k \\ \boldsymbol{0} \end{bmatrix} = -(\boldsymbol{g}_1)_k^T(\boldsymbol{H}_1)_k^{-1}(\boldsymbol{g}_1)_k < 0\,.$$

Furthermore, for the particular case of linear least squares problems (2.5), by denoting the residual vector with $\boldsymbol{r}$, i.e. $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$ and by partitioning $\boldsymbol{A}$ as previously, that is $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \\ \boldsymbol{A}_3 & \boldsymbol{A}_4 \end{bmatrix}$, we point out that the following form for the reduced gradient can be deducted

$$\boldsymbol{g}_k^R = -2\begin{bmatrix} (\boldsymbol{A}_1^T)_k \\ \boldsymbol{0} \end{bmatrix}\boldsymbol{r}_k\,.$$

Hence (2.7) can be rewritten as

$$\boldsymbol{p}_k^R = -\begin{bmatrix} 2(\boldsymbol{A}_1^T\boldsymbol{A}_1)_k & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{n-r} \end{bmatrix}^{-1}\begin{bmatrix} (\boldsymbol{g}_1)_k \\ \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} 2(\boldsymbol{A}_1^T\boldsymbol{A}_1)_k & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{n-r} \end{bmatrix}^{-1}\begin{bmatrix} 2(\boldsymbol{A}_1^T)_k \\ \boldsymbol{0} \end{bmatrix}\boldsymbol{r}_k\,.$$

By restricting the computation to the r free variables, $\boldsymbol{p}_k^R$ can be obtained through the following expressions:

$$(\boldsymbol{p}_1)_k = (\boldsymbol{A}_1^T\boldsymbol{A}_1)_k^{-1}(\boldsymbol{A}_1^T)_k\boldsymbol{r}_k \tag{2.8}$$

$$\boldsymbol{p}_k^R = \begin{bmatrix} (\boldsymbol{p}_1)_k \\ \boldsymbol{0} \end{bmatrix} \tag{2.9}$$

Equation (2.8) suggests a least squares formulation for $\boldsymbol{p}_1$ at each iteration, which is

$$\boldsymbol{p}_1 = \underset{\boldsymbol{s}}{\operatorname{argmin}}\,\|\boldsymbol{A}_1\boldsymbol{s} - \boldsymbol{r}\|_2\,.$$

The Reduced Projected-Newton method is detailed in the following algorithm:

Reduced Projected-Newton Algorithm to solve: $\min_{\boldsymbol{x}\geq 0} \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2$

choose initial $\boldsymbol{x}_0$

for $k = 0, 1, 2, \ldots$

$\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$

$(\boldsymbol{p}_1)_k = \operatorname{argmin}_{\boldsymbol{p}_1} \|\boldsymbol{A}_1\boldsymbol{p}_1 - \boldsymbol{r}\|_2$

$\boldsymbol{p}_k^R = \begin{bmatrix} (\boldsymbol{p}_1)_k \\ \boldsymbol{0} \end{bmatrix}$

Choose steplength τ_k

$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \tau_k \boldsymbol{p}_k^R$

end

The Projected-Newton direction in (2.6) can be rewritten by means of the previous partitioning as

$$\boldsymbol{p} = -(\boldsymbol{H}^R)^{-1}\boldsymbol{g} = -\begin{bmatrix} (\boldsymbol{H}_1)^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{n-r} \end{bmatrix} \begin{bmatrix} \boldsymbol{g}_1 \\ \boldsymbol{g}_2 \end{bmatrix} = -\begin{bmatrix} (\boldsymbol{H}_1)^{-1}\boldsymbol{g}_1 \\ \boldsymbol{g}_2 \end{bmatrix},$$

revealing that it differs from the reduced direction (2.7) just in the free components. In particular, since the $n-r$ free entries correspond to the $\boldsymbol{x}_k$ components in the active set, once that $(\boldsymbol{x}_k)_i$ has been projected, the proposed direction $\boldsymbol{p}_k^R$ will not move it away from zero. In general, this feature may reveal a stagnant behavior of the algorithm, but it is well suited to the particular case of blind deconvolution application.

Let us now focus on the imaging framework. In order to overcome the severe ill-posedness affecting such problems, we present here an algorithm with a regularizing effect. In particular, we insert Tikhonov regularization in the $\boldsymbol{p}_k^R$ computation step:

$$(\boldsymbol{p}_1^\lambda)_k = \operatorname*{argmin}_{\boldsymbol{p}_1} \left\| \begin{bmatrix} \boldsymbol{A}_1 \\ \lambda_k \boldsymbol{I} \end{bmatrix} \boldsymbol{p}_1 - \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{0} \end{bmatrix} \right\|_2$$

$$\boldsymbol{p}_k^R = \begin{bmatrix} (\boldsymbol{p}_1^\lambda)_k \\ \boldsymbol{0} \end{bmatrix}.$$

It's important to remark that in the imaging framework, when spatially invariant blurring is assumed, $\boldsymbol{A}$ is a block structured matrix; for instance, if periodic boundary

conditions are assumed, $\boldsymbol{A}$ is a block circulant matrix with circulant blocks.
Unfortunately, when considering the reduced matrix $\boldsymbol{A}_1$ the block structured feature is solely lost. We discuss how to overcome this difficulty in 2.3.2.

Due to its balancing role between the requests of fidelity to the data and smoothing assumption of the solution, the choice of an appropriate regularization parameter λ is crucial. Classical methods to estimate λ are the L-curve principle [32] and Generalized Cross Validation (GCV) [29, 30, 31, 55]. In the following sections we briefly describe the GCV method and propose an efficient way to compute the regularization parameter λ for large scale application.

2.3.1 Regularization through GCV

GCV is a statistical method whose basic idea is that a good λ should be able to predict missing data values. Hence by leaving out each component of the data in turn, we define the so-called GCV function

$$G(\lambda) = \frac{r\left\| \boldsymbol{r} - \boldsymbol{A}_1 \begin{bmatrix} \boldsymbol{A}_1 \\ \lambda \boldsymbol{I} \end{bmatrix}^{\dagger} \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{0} \end{bmatrix} \right\|_2^2}{\left(trace\left(\boldsymbol{I} - \boldsymbol{A}_1 \begin{bmatrix} \boldsymbol{A}_1 \\ \lambda \boldsymbol{I} \end{bmatrix}^{\dagger} \right) \right)^2} = \frac{r\left\| \boldsymbol{r} - \boldsymbol{A}_1 \boldsymbol{p}_1^{\lambda} \right\|_2^2}{\left(trace\left(\boldsymbol{I} - \boldsymbol{A}_1 \begin{bmatrix} \boldsymbol{A}_1 \\ \lambda \boldsymbol{I} \end{bmatrix}^{\dagger} \right) \right)^2} \tag{2.10}$$

which measures the prediction error. Since the regularized solution, defined by means of λ, is expected to well predict the missed value, we seek for the minimum of $G(\lambda)$. Equation (2.10) can be rewritten by means of the SVD of $\boldsymbol{A}_1 = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, where $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n]$, $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r]$ and $\boldsymbol{\Sigma}$ is a diagonal matrix with entries $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$:

$$G(\lambda) = \frac{r\left(\sum_{i=1}^{r} \frac{\lambda^2}{\sigma_i^2 + \lambda^2} \boldsymbol{u}_i^T \boldsymbol{r} \right) + \sum_{i=r+1}^{n} (\boldsymbol{u}_i^T \boldsymbol{r})^2}{\left((n-r) + \sum_{i=1}^{r} \frac{\lambda^2}{\sigma_i^2 + \lambda^2} \right)^2}. \tag{2.11}$$

Problems of small dimensions can be thus addressed to the following Regularized Reduced Projected-Newton algorithm:

Regularized Reduced Projected-Newton Algorithm to solve: $\min_{\boldsymbol{x}\geq 0} \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2$

choose initial $\boldsymbol{x}_0$

for $k = 0, 1, 2, \ldots$

$\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$

Compute λ_k by performing GCV as in (2.11)

$$(\boldsymbol{p}_1^{\lambda})_k = \underset{\boldsymbol{p}_1}{\operatorname{argmin}} \left\| \begin{bmatrix} \boldsymbol{A}_1 \\ \lambda_k \boldsymbol{I} \end{bmatrix} \boldsymbol{p}_1 - \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{0} \end{bmatrix} \right\|_2$$

$$\boldsymbol{p}_k^R = \begin{bmatrix} (\boldsymbol{p}_1^{\lambda})_k \\ \boldsymbol{0} \end{bmatrix}$$

Choose steplength τ_k

$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \tau_k \boldsymbol{p}_k^R$

end

2.3.2 Lanczos bidiagonalization

For large-scale applications, like image deblurring, computing the SVD of $\boldsymbol{A}_1$ at each iteration of the projected method is computationally very expensive. A common way to overcome this situation consists of approaching the least squares problem

$$\min_{\boldsymbol{p}_1} \left\| \begin{bmatrix} \boldsymbol{A}_1 \\ \lambda \boldsymbol{I} \end{bmatrix} \boldsymbol{p}_1 - \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{0} \end{bmatrix} \right\|_2^2 \tag{2.12}$$

through an iterative procedure based on Lanczos bidiagonalization (LBD). Given $\boldsymbol{A}_1 \in \mathbb{R}^{n\times r}$ and $\boldsymbol{r} \in \mathbb{R}^n$, the j-th iteration of Lanczos bidiagonalization produces a $n \times (j+1)$ matrix $\boldsymbol{W}_j$, a $r \times j$ matrix $\boldsymbol{Y}_j$, an r-vector $\boldsymbol{t}_{j+1}$ and $(j+1) \times j$ bidiagonal matrix $\boldsymbol{B}_j$ such that

$$\boldsymbol{A}_1^T \boldsymbol{W}_j = \boldsymbol{Y}_j \boldsymbol{B}_j^T + \gamma_{j+1} \boldsymbol{t}_{j+1} \boldsymbol{e}_{j+1}^T \tag{2.13}$$

$$\boldsymbol{A}_1 \boldsymbol{Y}_j = \boldsymbol{W}_j \boldsymbol{B}_j \,, \tag{2.14}$$

where $\boldsymbol{e}_{j+1}$ is the $(j+1)$-st vector of the canonical basis and $\boldsymbol{B}_j$ is

$$\boldsymbol{B}_j = \begin{bmatrix} \gamma_1 & & & \\ \beta_2 & \gamma_2 & & \\ & \ddots & \ddots & \\ & & \beta_j & \gamma_j \\ & & & \beta_{j+1} \end{bmatrix}.$$

An interesting property of $\boldsymbol{B}_j$ is that its singular values are a good approximation of certain singular values of the initial matrix $\boldsymbol{A}_1$. In particular, when $\boldsymbol{A}_1$ derives from the discretization of ill-posed problem, the singular values of $\boldsymbol{B}_j$ generated at the first iterations of the bidiagonalization process approximate really well the largest σ_i. This special feature allows to replace, during the evaluation of the GCV function (2.11), the SVD decomposition of $\boldsymbol{A}_1$ with that of $\boldsymbol{B}_j$, which is definitely cheaper.

Hence at each iteration j of Lanczos bidiagonalization, the least squares problem (2.12) is replaced with

$$\min_{\widetilde{\boldsymbol{p}_1}} \left\| \begin{bmatrix} \boldsymbol{B}_j \\ \lambda_j \boldsymbol{I} \end{bmatrix} \widetilde{\boldsymbol{p}_1} - \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{0} \end{bmatrix} \right\|_2^2,$$

where λ_j is chosen by evaluating the so-rewritten GCV function:

$$\begin{aligned} \widetilde{G}(\lambda) &= \frac{r \left\| \boldsymbol{r} - \boldsymbol{B}_j \begin{bmatrix} \boldsymbol{B}_j \\ \lambda \boldsymbol{I} \end{bmatrix}^{\dagger} \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{0} \end{bmatrix} \right\|_2^2}{\left(trace \left(\boldsymbol{I} - \boldsymbol{B}_j \begin{bmatrix} \boldsymbol{B}_j \\ \lambda \boldsymbol{I} \end{bmatrix}^{\dagger} \right) \right)^2} = \\ &= \frac{r \left\| \boldsymbol{r} - \boldsymbol{B}_j \widetilde{\boldsymbol{p}_1} \right\|_2^2}{\left(trace \left(\boldsymbol{I} - \boldsymbol{B}_j \begin{bmatrix} \boldsymbol{B}_j \\ \lambda \boldsymbol{I} \end{bmatrix}^{\dagger} \right) \right)^2} \end{aligned} \tag{2.15}$$

We remark that $\boldsymbol{B}_j$ is of really small dimension, even in the large scale case, thus the computation of $\boldsymbol{B}_j^{\dagger}$ through its SVD is really cheap. For the stopping criteria of Lanczos bidiagonalization procedure, we refer to [17].

2.3.3 A LBD-Regularized Reduced Projected-Newton method

Now that we have gathered all of the necessary tools, we describe in detail the proposed projection method for linear least squares subject to nonnegative constraints. The descent direction is computed through an Hessian approximation $\boldsymbol{A}_1^T \boldsymbol{A}_1$ generated by

restricting to the free variables; Tikhonov regularization is added, whose regularization parameter is estimated through GCV. Moreover, due to the large scale, the GCV function is evaluated through the singular values of the matrix bidiagonalizing $\boldsymbol{A}_1^T\boldsymbol{A}_1$. We summarize the proposed method in the following algorithm.

LBD-Regularized Reduced PN Algorithm to solve: $\min_{\boldsymbol{x}\geq 0} \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2$

choose initial $\boldsymbol{x}_0$

for $k = 0, 1, 2, \ldots$

$\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$

for $j = 1, 2, \ldots$

Compute $\boldsymbol{B}_j$ through (2.13)

Compute λ_j by evaluating (2.15)

$$(\widetilde{\boldsymbol{p}_1})_k = \underset{\boldsymbol{p}_1}{\operatorname{argmin}} \left\| \begin{bmatrix} \boldsymbol{B}_j \\ \lambda_j \boldsymbol{I} \end{bmatrix} \boldsymbol{p}_1 - \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{0} \end{bmatrix} \right\|_2$$

end

$$\boldsymbol{p}_k^R = \begin{bmatrix} (\widetilde{\boldsymbol{p}_1})_k \\ \boldsymbol{0} \end{bmatrix}$$

Choose steplength τ_k

$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \tau_k \boldsymbol{p}_k^R$

end

2.4 Numerical experiment

This section contains some numerical results achieved by the proposed Reduced Projected-Newton algorithm when nonnegative constraints are imposed. In particular we distinguish between problems of small and large dimension; for the first case we applied the algorithm referred to as Regularized Reduced Projected-Newton (RR-PN), while for large scale we performed the LBD-Regularized Reduced Projected-Newton (LBD-RR-PN).
The steplength τ_k has been chosen through the Cauchy rule, that is

$$\tau_k = \underset{\tau}{\operatorname{argmin}}\, h(\boldsymbol{x}_k + \tau \boldsymbol{p}_k^R)\,;$$

in addiction, a line search on the arc has been performed in order to ensure convergence [8]. The iterative scheme has been stopped when one of the following criteria has been

reached:

- the residual satisfies $\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k\|_2 \leq \|\boldsymbol{b}\|_2\, \delta_1$;
- the normal equation residual satisfies $\|\boldsymbol{A}^T\boldsymbol{b} - \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}_k\|_2 \leq \|\boldsymbol{A}^T\boldsymbol{b}\|_2\, \delta_2$;
- a maximum numerical `MaxLS` of line search reductions has been reached;
- a maximum numerical `MaxIter` of iterations has been reached.

2.4.1 Small case examples

We report here simple numerical experiments showing the effectiveness of the proposed technique in comparison with non-regularized Gradient-Projection algorithm. The evaluation of the results will be carried out through the test problems presented in the *Regularization Tools* [28]:

Phillips is the discretization of a Fredholm integral equation of the first kind

$$b(s) = \int_{-6}^{6} a(s,t)x(t)\mathtt{d}t$$

where the kernel is

$$a(s,t) = \begin{cases} 1 + \cos\left(\frac{\pi}{3}(s-t)\right) & \text{if } |s-t| < 3 \\ 0 & \text{if } |s-t| \geq 3 \end{cases}$$

$$x(t) = \begin{cases} 1 + \cos\left(\frac{\pi}{3}t\right) & \text{if } |t| < 3 \\ 0 & \text{if } |t| \geq 3 \end{cases},$$

$$b(s) = (6 - |s|)\left(1 + \frac{1}{2}\cos\left(\frac{\pi}{3}s\right)\right) + \frac{9}{2\pi}\sin\left(\frac{\pi|s|}{3}\right).$$

Among the possible choices, we choose $x(t) = t$ and $b(s) = \frac{1}{6}(s^3 - s)$.

Shaw is a one-dimensional image restoration problem deriving from the discretization of a Fredholm integral equation of the first kind, where the kernel is

$$a(s,t) = (\cos(s) + \cos(t))^2 \frac{\sin^2(u)}{u^2}, \quad u(s,t) = \pi(\sin(s) + \sin(t))$$

and the solution is

$$x(t) = 2\exp\left(-6(t-0.8)^2\right) + \exp\left(-2(t+0.5)^2\right),$$

with $s, t \in [-\pi/2, \pi/2]$.

Deriv2 is the discretization of the Fredholm integral equation of the first kind

$$b(s) = \int_0^1 a(s,t)x(t)\mathtt{d}t$$

where $s \in [0,1]$, the kernel is

$$a(s,t) = \begin{cases} s(t-1) & \text{if } s < t \\ t(s-1) & \text{if } s \geq t \end{cases} .$$

Among the possible choices, we choose $x(t) = t$ and $b(s) = \frac{1}{6}(s^3 - s)$.

Baart is the discretization of the Fredholm integral equation of the first kind

$$b(s) = \int_0^\pi a(s,t)x(t)\mathtt{d}t$$

where $s \in [0,\pi]$, the kernel is $a(s,t) = \exp(s\cos(t))$, the solution is $x(t) = \sin(t)$ and the right-hand side is $b(s) = 2\frac{\sinh(s)}{s}$.

Heat is an inverse heat equation using the Volterra integral equation of the first kind with kernel $a(s,t) = k(s-t)$, where

$$k(t) = \frac{t^{-3/2}}{2\sqrt(\pi)} \exp(-1/4t) \quad t \in [0,1]\,.$$

Gravity is a one-dimensional gravity surveying model problem deriving from the discretization of a Fredholm integral equation of the first kind, where the kernel is

$$a(s,t) = \frac{1}{4}\left(\frac{1}{16} + (s-t)^2\right)^{-3/2}$$

and the solution we use is

$$x(t) = \sin(\pi t) + 0.5\sin(2\pi t)\,,$$

with $t \in [0,1]$.

For each test problem, the data can be constructed with the simple MATLAB statement: `[A, b_true, x_true] = TestProblem(n);` where is not specified, $\boldsymbol{b}_{true}$ is obtained as $\boldsymbol{b}_{true} = \boldsymbol{A}\boldsymbol{x}_{true}$. Moreover, we assume $n = 256$ and Gaussian noise with zero mean and variance equal to 1 and scaled so that $\frac{\|\boldsymbol{\eta}\|}{\|\boldsymbol{A}\boldsymbol{x}\|} = 0.1$; we set $\delta_1 = 10^{-6}$, $\delta_2 = 10^{-6}$, `MaxLS` $= 20$ and `MaxIter` $= 1000$. Figure 2.3 contains the reconstruction provided by the proposed Regularized Reduced Projected-Newton; compared with a Gradient-Projection method without any regularization, the proposed algorithm seems to avoid the classical semiconvergent behavior due to ill-posedness (Figure 2.4).

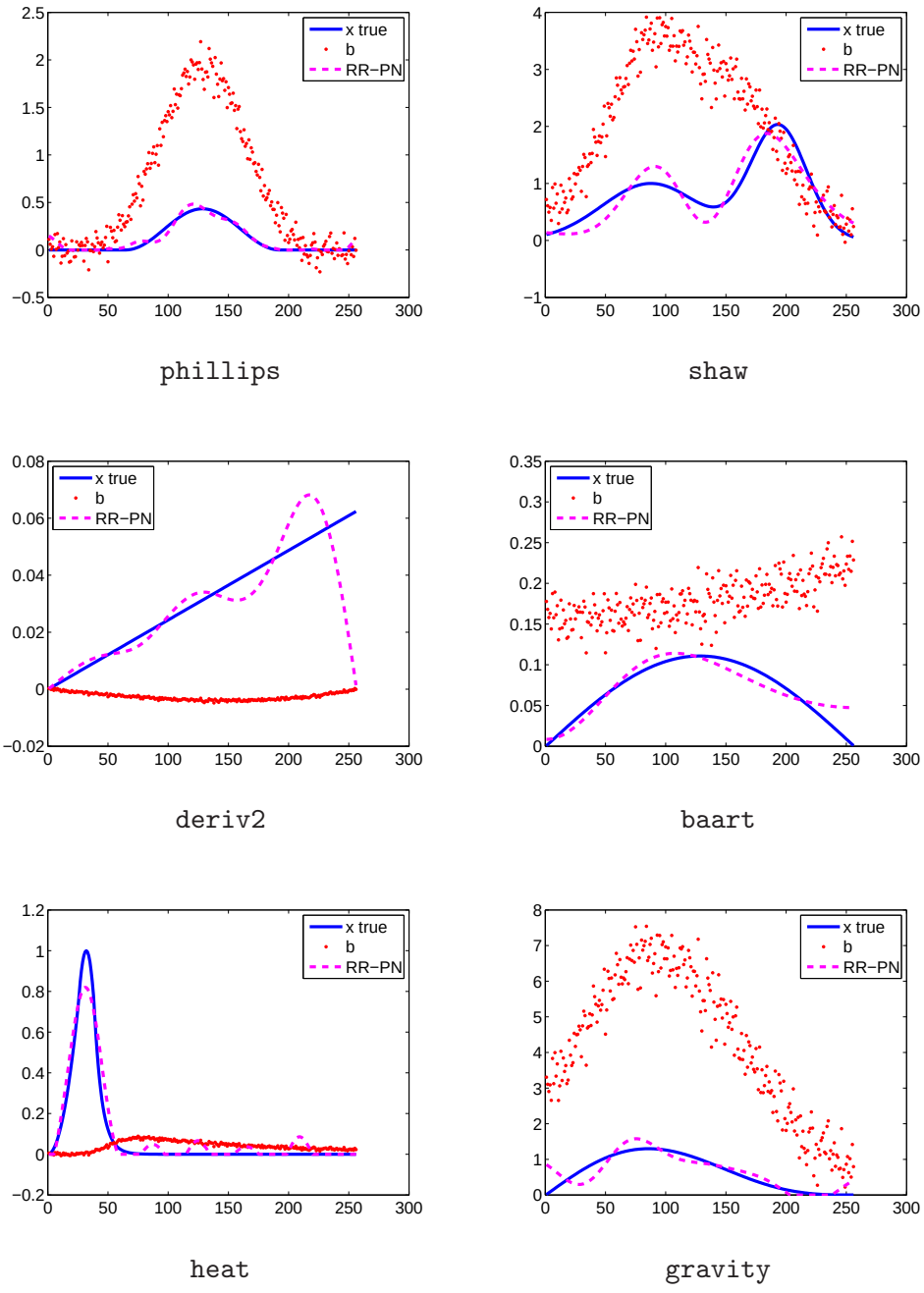


Figure 2.3: Reconstructions of the solution provided by the Regularized Reduced Projected-Newton method (RR-PN, magenta dashed), compared with the true object (blue solid) and the noisy data (red dashdot).

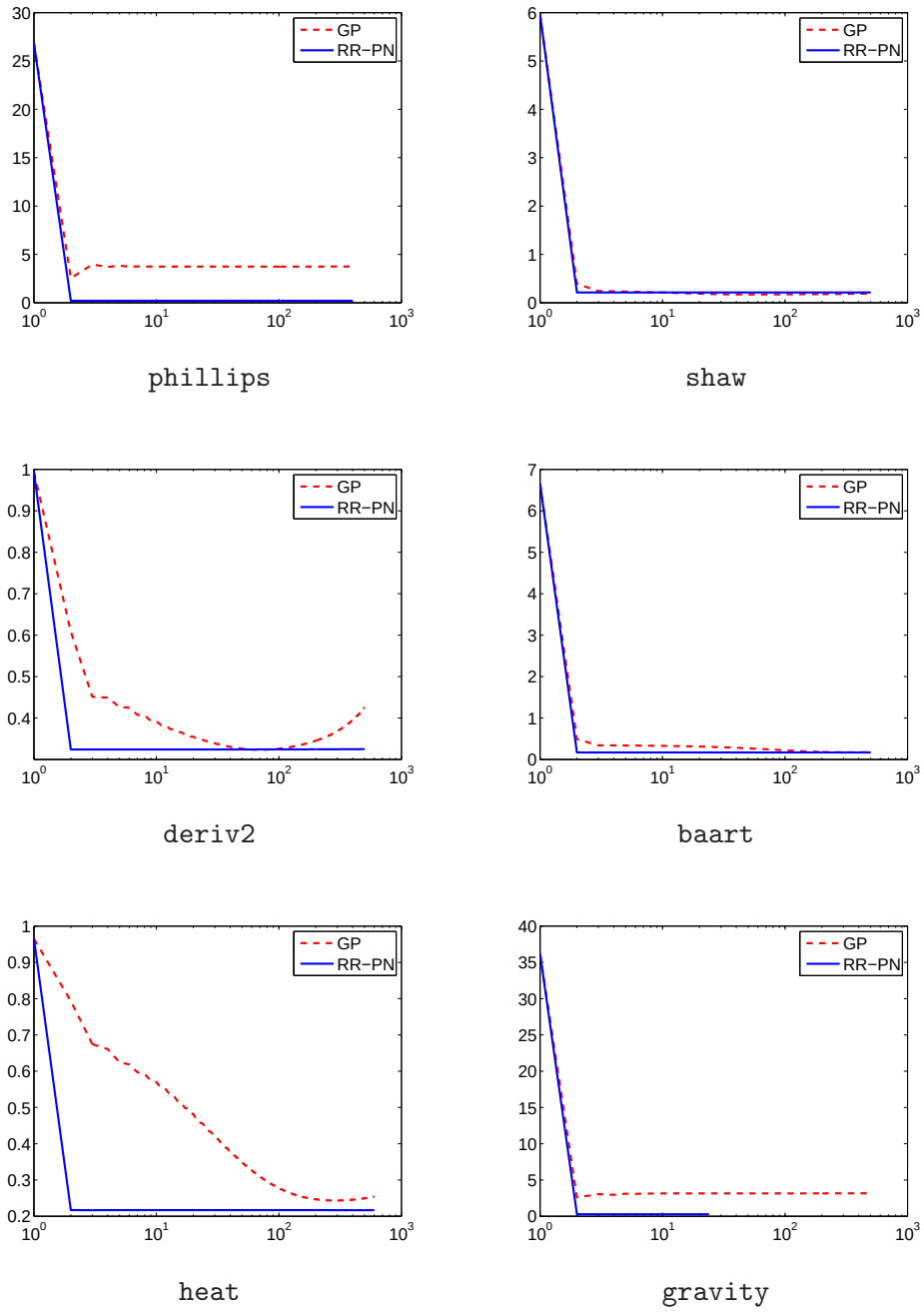


Figure 2.4: Relative errors achieved by non-regularized Gradient-Projection (GP, red dashed) and the Regularized Reduced Projected-Newton method (RR-PN, blue solid).

2.4.2 Deblurring application

The large-scale problem comes from the *Restore Tools* iterative image deblurring package and we consider the satellite image (256 × 256 pixel). For the implementation of both the GCV method and Lanczos bidiagonalization, we refer to *HyBR tool*[2]. In particular, for the large scale case, a specific implementation of the matrix-vector multiplication involving $\boldsymbol{AD}$ is necessary in order to exploit the particular structure of the matrix. The blurred image $\boldsymbol{b}_{true}$ has been corrupted by 10% Gaussian noise with zero mean and variance equal to 1. Moreover we set $\delta_1 = 10^{-6}$, $\delta_2 = 10^{-6}$, `MaxLS` = 20 and `MaxIter` = 1000. In addition, we introduce a further criteria that stops the iterative procedure when $\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_2 \leq 10^{-6}$.

Figure 2.5 contains the relative errors of the Gradient-Projection method without any regularization and the proposed LBD-Regularized Reduced Projected-Newton. The minimum errors achieved are, respectively, 0.399436 (iteration 624) and 0.407314 (iteration 161).

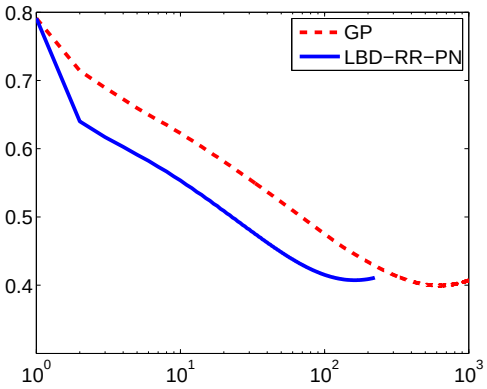


Figure 2.5: Relative errors achieved by non-regularized Gradient-Projection (GP, red dashed) and the LBD-Regularized Reduced Projected-Newton method (LBD-RR-PN, blue solid).

[2]http://www.math.vt.edu/people/jmchung/hybr.html

Chapter 3

Blind deconvolution as separable problem

3.1 Blind deconvolution model problem

In order to model the image formation process, let us consider the following convolution equation,

$$b(s,t) = \mathcal{K}(s,t) * x(s,t) + \eta(s,t)\,,$$

where $x(s,t)$ is a function representing the true image, which is convolved with a point spread function (PSF), $\mathcal{K}(s,t)$, and after including $\eta(s,t)$, to model additive noise, we obtain the observed image, $b(s,t)$. Although we could use a general integral equation to model the image formation process, because data are given in discrete form, we prefer to move directly to the discretized equation

$$\boldsymbol{b} = \boldsymbol{\mathcal{K}} * \boldsymbol{x} + \boldsymbol{\eta}\,,$$

where

- $\boldsymbol{b}$ is a vector representing the observed, blurred and noisy image;
- $\boldsymbol{x}$ is vector representing the unknown true image we wish to reconstruct;
- $\boldsymbol{\mathcal{K}}$ is the PSF;
- $\boldsymbol{\eta}$ is a vector that represents unknown additive noise in the measured data. Generally $\boldsymbol{\eta}$ is a combination of background and readout noise, where the background noise is modeled as a Poisson random process with fixed Poisson parameter, and the readout noise is modeled as a Gaussian random process with zero mean and fixed variance [1, 2, 53, 54].

Blind deconvolution occurs when, given only the blurred noisy image, one attempts to recover both the PSF and the true image. There are many existing algorithms to solve blind deconvolution, based on the alternating minimization over the PSF $\boldsymbol{\mathcal{K}}$ and the image $\boldsymbol{x}$ [14, 15, 57, 58], resulting in the optimization problem

$$\min_{\boldsymbol{\mathcal{K}},\boldsymbol{x}} \left\| \boldsymbol{\mathcal{K}} * \boldsymbol{x} - \boldsymbol{b} \right\|_{\mathcal{L}^2}^2 . \tag{3.1}$$

This idea corresponds to the procedure underlying Alternating Optimization, thus the following scheme is built:

$$\begin{aligned} \boldsymbol{\mathcal{K}}_\ell &= \operatorname*{argmin}_{\boldsymbol{\mathcal{K}}} \left\| \boldsymbol{\mathcal{K}} * \boldsymbol{x}_{\ell-1} \right\|_{\mathcal{L}^2}^2 \\ \boldsymbol{x}_\ell &= \operatorname*{argmin}_{\boldsymbol{y}} \left\| \boldsymbol{\mathcal{K}}_\ell * \boldsymbol{x} \right\|_{\mathcal{L}^2}^2 \end{aligned} \tag{3.2}$$

As the blind deconvolution problem is ill-posed with respect to both the image and the blurring function, regularization on both $\boldsymbol{\mathcal{K}}$ and $\boldsymbol{x}$ must be applied, thus (3.1) becomes

$$\min_{\boldsymbol{\mathcal{K}},\boldsymbol{x}} \left\| \boldsymbol{\mathcal{K}} * \boldsymbol{x} - \boldsymbol{b} \right\|_{\mathcal{L}^2}^2 + \lambda_1 \mathcal{R}_1(\boldsymbol{\mathcal{K}}) + \lambda_2 \mathcal{R}_2(\boldsymbol{x}) , \tag{3.3}$$

with the related alternating scheme. Different choices of regularization functionals $\mathcal{R}_1$ and $\mathcal{R}_2$ can be considered, such as Total Variation [48, 49]. In [15] the convergence of the Alternating Optimization for blind deconvolution has been proved; in [14] Total Variation blind deconvolution has been addressed.
In general, in any blind deconvolution problem it is necessary to impose some assumptions about the blur and the image, because the minimization problem (3.3) may not have a unique solution thus it does not always yield a meaningful solution. Common assumptions are the nonnegativity on $\boldsymbol{\mathcal{K}}$ and $\boldsymbol{x}$ or the flux of the PSF to be 1.
In general, the PSF is represented by a matrix $\boldsymbol{A}$, typically sparse and/or structured.

In this work we make the assumption that the blurring operator can be described through a model function depending on unknown parameters, i.e. $\boldsymbol{A} = \boldsymbol{A}(\boldsymbol{y})$. Such problems are sometimes referred to as *semi-blind* or *myopic deconvolution*. The resulting blind deconvolution model is an inverse problem of the form

$$\boldsymbol{b} = \boldsymbol{A}(\boldsymbol{y}_{\text{true}})\boldsymbol{x}_{\text{true}} + \boldsymbol{\eta} \tag{3.4}$$

where $\boldsymbol{b} \in \mathbb{R}^n$ is the measured, blurred and noisy image, $\boldsymbol{\eta} \in \mathbb{R}^n$ models the noise and $\boldsymbol{x}_{\text{true}} \in \mathbb{R}^n$ represents the unknown true image. The vector $\boldsymbol{y}_{\text{true}} \in \mathbb{R}^m$ is unknown and $\boldsymbol{A}(\boldsymbol{y})$ is a nonlinear operator modeling the blurring, which maps $\boldsymbol{y}$ into an $n \times n$ matrix. The matrix $\boldsymbol{A}(\boldsymbol{y})$ is typically severely ill-conditioned, with singular values that cluster at zero.

Equation (3.4) is a *separable* inverse problem because the measured data depend linearly on the unknown vector $\boldsymbol{x}$ and nonlinearly on the unknown vector $\boldsymbol{y}$. In addition to blind deconvolution, separable inverse problems arise in many applications, such as super-resolution (which is an example of image data fusion) [16, 35], cryo-EM microscopy imaging [18, 24, 34, 39, 45, 51], and in seismic imaging applications [27].

Given the data $\boldsymbol{b}$ and the mapping $\boldsymbol{A}(\cdot)$, the aim is to compute approximations $\boldsymbol{x}$ and $\boldsymbol{y}$ of, respectively, $\boldsymbol{x}_{\text{true}}$ and $\boldsymbol{y}_{\text{true}}$. Generally this is done by defining an objective function, $f(\boldsymbol{x}, \boldsymbol{y})$, and then using an optimization algorithm to solve the separable problem

$$\min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y})\,, \tag{3.5}$$

that will have different expressions for Variable Projection and Alternating Optimization, depending on the chosen regularization approach. Indeed, as already mentioned in section 1.5, in Variable Projection the linear solution $\boldsymbol{x}$ must necessarily be computed exactly. By this we mean that to overcome ill-posedness a direct regularization approach must be used. On the other hand, Alternating Optimization does not have such accuracy request, thus iterative regularization is preferable in terms of computational cost.

We remark that in many applications, especially in image processing, a nonnegativity constraint on $\boldsymbol{x}$ is used to obtain physically meaningful solutions (i.e., when $\boldsymbol{x}$ contains pixels intensity values, these should be nonnegative). Since our interest concerns the nonnegative formulation, we state blind deconvolution as the following nonnegative separable least squares problem

$$\min_{\boldsymbol{x},\boldsymbol{y}} \|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x} - \boldsymbol{b}\|_2^2 \qquad \text{subject to} \quad \boldsymbol{x} \geq 0\,. \tag{3.6}$$

3.1.1 Variable Projection

By applying the Variable Projection method to solve the nonlinear problem (3.6), a reduced cost functional is obtained

$$\psi(\boldsymbol{y}) = \|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x}(\boldsymbol{y}) - \boldsymbol{b}\|_2^2\,, \tag{3.7}$$

which depends only on $\boldsymbol{y}$, with

$$\boldsymbol{x}(\boldsymbol{y}) = \operatorname*{argmin}_{\boldsymbol{x}\geq 0} \|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x} - \boldsymbol{b}\|_2\,. \tag{3.8}$$

In order to solve the previous nonnegative linear least squares, we apply the LBD-Regularized Projected-Newton algorithm proposed in section 2.3.

In general, computing a minimum of ψ is difficult because it may not have a well defined global minimum, and there are many local minima in which an iterative solver,

such as a standard Gauss-Newton method [38, 43], can become trapped. In the following we discuss the effectiveness of imposing nonnegative constraints in order to obtain a smoother objective function.

3.1.1.1 Objective function investigations

Our aim is now to explain why the results obtained with the nonnegative model are better than the results of the unconstrained formulation. In order to face ill-posedness, we add Tikhonov regularization to both formulations. By this we mean that we compare between the regularized nonnegative problem

$$\min_{\boldsymbol{x},\boldsymbol{y}} \psi_0^{\beta_1} = \left\| \begin{bmatrix} \boldsymbol{A}(\boldsymbol{y}) \\ \beta_1 \boldsymbol{I} \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} \right\|_2^2 \qquad \text{subject to} \quad \boldsymbol{x} \geq \boldsymbol{0} \tag{3.9}$$

and the regularized unconstrained problem

$$\min_{\boldsymbol{x},\boldsymbol{y}} \varphi_0^{\beta^2} = \left\| \begin{bmatrix} \boldsymbol{A}(\boldsymbol{y}) \\ \beta_2 \boldsymbol{I} \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} \right\|_2^2 . \tag{3.10}$$

Once applied the Variable Projection, we make a comparison between the nonnegative objective function deriving from (3.9)

$$\psi^{\beta_1}(\boldsymbol{y}) = \|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x}(\boldsymbol{y}) - \boldsymbol{b}\|_2^2 , \tag{3.11}$$

with

$$\boldsymbol{x}(\boldsymbol{y}) = \underset{\boldsymbol{x}\geq 0}{\operatorname{argmin}} \left\| \begin{bmatrix} \boldsymbol{A}(\boldsymbol{y}) \\ \beta_1 \boldsymbol{I} \end{bmatrix} - \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} \right\|_2 . \tag{3.12}$$

and the unconstrained one, deriving from (3.10),

$$\varphi^{\beta_2}(\boldsymbol{y}) = \|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x}(\boldsymbol{y}) - \boldsymbol{b}\|_2^2 , \tag{3.13}$$

with

$$\boldsymbol{x}(\boldsymbol{y}) = \underset{\boldsymbol{x}}{\operatorname{argmin}} \left\| \begin{bmatrix} \boldsymbol{A}(\boldsymbol{y}) \\ \beta_2 \boldsymbol{I} \end{bmatrix} - \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} \right\|_2 . \tag{3.14}$$

We consider a simple example of Gaussian PSF defined by

$$\mathcal{K}(s,t) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{1}{2} \begin{bmatrix} s & t \end{bmatrix} \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}^{-1} \begin{bmatrix} s \\ t \end{bmatrix} \right) . \tag{3.15}$$

In this case the nonlinear parameter vector is $\boldsymbol{y} = (\sigma, \sigma, 0) = (2, 2, 0)$, that is, the objective function depends on only one parameter. We compare the objective functions

$\psi^{\beta_1}(\boldsymbol{y})$ in (3.11) and $\varphi^{\beta_2}(\boldsymbol{y})$ in (3.13), recalling that the only difference between them is the computation of the solution $\boldsymbol{x}(\boldsymbol{y})$ of the deconvolution step.

We first evaluate $\varphi^{\beta_2}(\boldsymbol{y})$ for various values of $\boldsymbol{y}$ and we obtain the plot shown in figure 3.1. For this example we see that the objective function does not have a minimum at

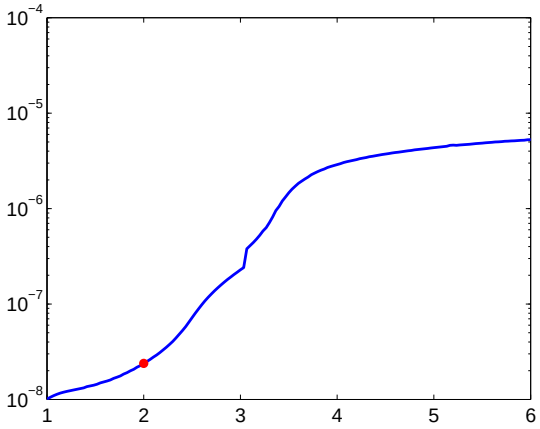


Figure 3.1: Plot of the objective function $\varphi^{\beta_2}(\boldsymbol{y})$. The regularization parameter needed to compute $\boldsymbol{x}(\boldsymbol{y})$ is chosen using a weighted generalized cross validation method. The red dot indicates $f(\boldsymbol{y}_{\text{true}})$.

the desired solution $\boldsymbol{y}_{\text{true}} = 2$. We next investigate how the accuracy of computing $\boldsymbol{x}(\boldsymbol{y})$ in our deconvolution step affects the behavior of the objective function. Recall that to obtain the objective function shown in Figure 3.1 we used Tikhonov regularization to compute $\boldsymbol{x}(\boldsymbol{y})$, using a weighted GCV scheme to choose regularization parameters [17]. It is possible that an alternative parameter choice method will result in an objective function with a well defined global minimum. However, rather than testing the plethora of other parameter choice methods, we instead find the "optimal" β_2 that results in an $\boldsymbol{x}(\boldsymbol{y})$ that minimizes the error,

$$\frac{\|\boldsymbol{x}(\boldsymbol{y}) - \boldsymbol{x}_{\text{true}}\|_2}{\|\boldsymbol{x}_{\text{true}}\|_2}. \tag{3.16}$$

Of course this is not possible in a realistic situation when the true object is not known, but it does provide us with an idea of what is potentially possible. Using this approach to find the "optimal" β_2 at each iteration, the objective function for various values of $\boldsymbol{y}$ is shown in Figure 3.2.

In this case, the objective function has a global minimum. But unfortunately the value of $\boldsymbol{y}$ at which the objective function reaches its global minimum is not the desired

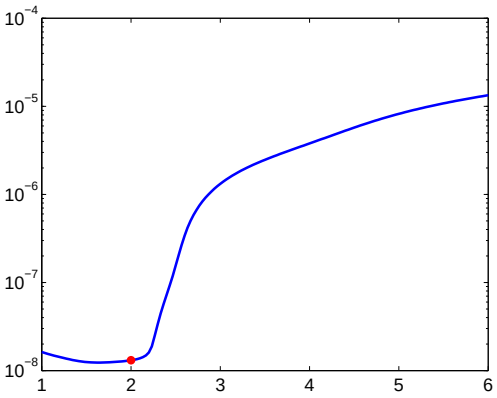


Figure 3.2: Plot of the objective function $\varphi^{\beta_2}(\boldsymbol{y})$. The regularization parameter needed to compute $\boldsymbol{x}(\boldsymbol{y})$ is chosen to minimize $\|\boldsymbol{x}(\boldsymbol{y}) - \boldsymbol{x}_{\text{true}}\|_2/\|\boldsymbol{x}_{\text{true}}\|_2$. The red dot indicates $\varphi^{\beta_2}(\boldsymbol{y}_{\text{true}})$.

value $\boldsymbol{y}_{\text{true}}$. Moreover, the objective function is very flat near the global minimum, which can make it difficult to recognize. We could go one step further and suppose we are lucky to have a deconvolution method that always computed the exact solution, $\boldsymbol{x}(\boldsymbol{y}) = \boldsymbol{x}_{\text{true}}$. In this extremely unrealistic case, we obtain the objective function shown in Figure 3.3.

Although it is impossible to have a deconvolution solver that always computes $\boldsymbol{x}(\boldsymbol{y}) = \boldsymbol{x}_{\text{true}}$, this, along with the case when we computed $\boldsymbol{x}(\boldsymbol{y})$ with an optimal regularization parameter, does suggest that we can help the optimization method by using a deconvolution solver that computes more physically realistic approximations of $\boldsymbol{x}_{\text{true}}$. One simple example to do this is to include a nonnegativity constraint within the deconvolution solver and compute the objective function $\psi^{\beta_1}(\boldsymbol{y})$ instead of $\varphi^{\beta_2}(\boldsymbol{y})$.

To test the constrained solver, we begin by using the optimal regularization parameters for the unconstrained problem, and compare them with more appropriate values for the constrained problem, found through experimentation[1]. The results are shown in Figure 3.4.

We observe that the nonnegativity constraint, along with well chosen regularization parameters, is very effective in producing an objective function $\psi^{\beta_1}(\boldsymbol{y})$ whose global minimum is near $\psi^{\beta_1}(\boldsymbol{y}_{\text{true}})$. However, we note that $\psi^{\beta_1}(\boldsymbol{y})$ has several local minima, and the optimization algorithms should avoid to be trapped in one of these, believing a global optimal solution has already been reached.

[1]We started with $\beta_1 = \beta_2^{opt}$ and reduced them systematically, eventually by a factor of 200, until we observed values that produced an objective function whose global minimum occurs near $\boldsymbol{y}_{\text{true}}$.

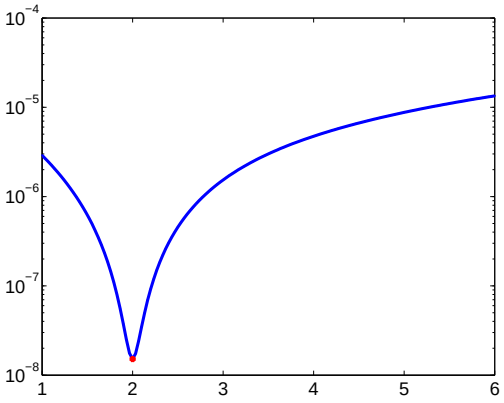
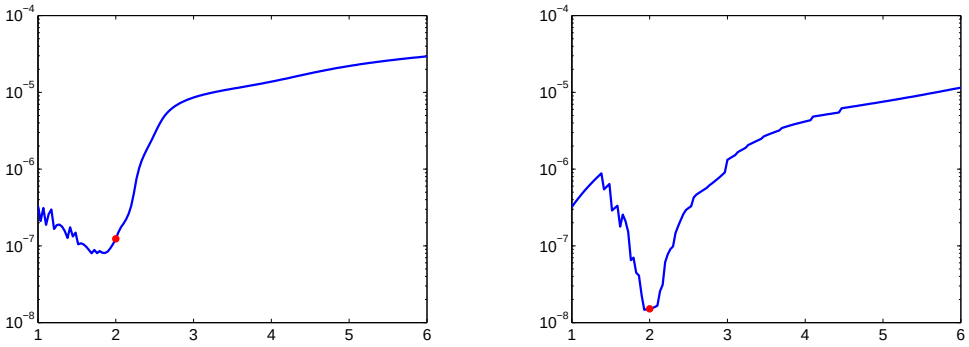


Figure 3.3: Plot of the objective function $\varphi^{\beta_2}(\boldsymbol{y})$. Here we use $\boldsymbol{x}(\boldsymbol{y}) = \boldsymbol{x}_{\text{true}}$. The red dot indicates $\varphi^{\beta_2}(\boldsymbol{y}_{\text{true}})$.



$\beta_1 = \beta_2^{opt}$ $\beta_1 = \beta_2^{opt}/200$

Figure 3.4: Plot of the objective function $\psi^{\beta_1}(\boldsymbol{y})$, where $\boldsymbol{x}(\boldsymbol{y})$ is obtained by solving, to high accuracy, the nonnegative constrained Tikhonov regularized deconvolution problem. The red dot indicates $\psi^{\beta_1}(\boldsymbol{y}_{\text{true}})$. The plot of $\psi^{\beta_1}(\boldsymbol{y})$ on the left was obtained by using $\beta_1 = \beta_2^{opt}$, the optimal β_2's for the unconstrained problem (see Figure 3.2), and the plot on the right was obtained using $\beta_1 = \beta_2^{opt}/200$.

3.1.1.2 Importance of the nonnegative formulation

We illustrate here how imposing nonnegative constraints on $\boldsymbol{x}$ provides better results for both the linear and nonlinear variables. Hence we compare equations (3.11) and (3.13).

We begin by using Tikhonov regularization in the deconvolution problem for the computation of $\boldsymbol{x}(\boldsymbol{y})$ in (3.14). The first consideration is to determine what to use for the regularization parameter β_2. Since the "optimal" value depends not only on the data $\boldsymbol{b}$ but also on the matrix $\boldsymbol{A}(\boldsymbol{y})$, it makes sense to use different values for different $\boldsymbol{y}$. It would be difficult to specify these values a priori, and thus it is appropriate to attempt to use a regularization parameter choice method, such as weighted generalized cross validation (WGCV) [17].

Then we consider including a nonnegativity constraint in the deconvolution; that is, we compute $\boldsymbol{x}(\boldsymbol{y})$ as in (3.12). To solve this constrained deconvolution problem we use the gradient projection method proposed in [5]. Although methods have been developed to select regularization parameters for constrained problems directly from the data (see, e.g., [3]), they are less developed than in the unconstrained case. In our tests, the regularization parameter has been selected heuristically.

We consider the Gaussian PSF, defined as

$$\mathcal{K}(s,t) = \frac{1}{2\pi\sqrt{\delta}} \exp\left(-\frac{1}{2} \begin{bmatrix} s & t \end{bmatrix} \begin{bmatrix} \sigma_1^2 & \rho^2 \\ \rho^2 & \sigma_2^2 \end{bmatrix}^{-1} \begin{bmatrix} s \\ t \end{bmatrix} \right) \tag{3.17}$$

where $\delta = \sigma_1^2\sigma_2^2 - \rho^4 > 0$. Moreover we consider as a true object the satellite image from the *Restore Tools*, blurred by the Gaussian PSF (3.17) defined by the parameter $\boldsymbol{y}_{true} = (\sigma_1, \sigma_2, \rho) = (1.5, 2, 0.5)$ and corrupted by 5% white Gaussian noise.

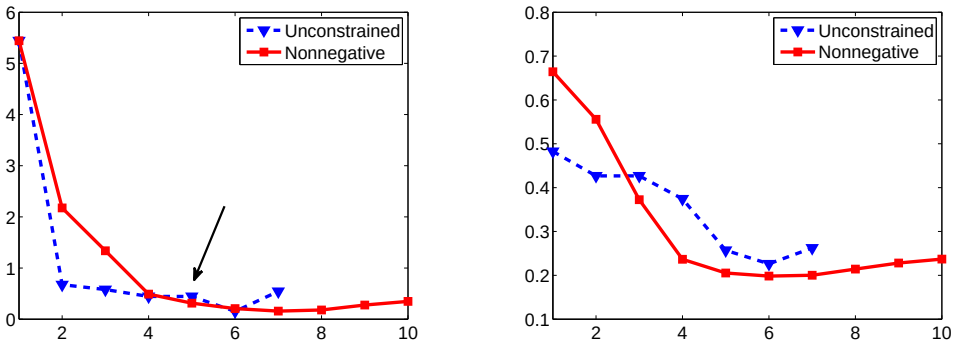
In figure 3.5 we compare the relative errors obtained by computing, in the blind deconvolution process, $\boldsymbol{x}(\boldsymbol{y})$ as the nonnegative solution (3.12) (solid line, red squares) and the unconstrained solution (3.14) (dashed line, blue triangles).

From these results we see that imposing nonnegative constraints allows to better eliminate the noise, especially in the black background; see figure 3.7, where we display surface mesh plots of a portion of the reconstructed images, compared with the truth. From the figures, it is evident that the constrained approach gives better approximations of both the $\boldsymbol{x}$ and $\boldsymbol{y}$ parameters. The left plot of figure 3.5 shows that the $\boldsymbol{y}$ error curve of the Gauss-Newton method has a semiconvergent behavior in both the cases, but it is flatter when imposing the nonnegative constraints. When a good solution is computed, the objective function becomes flat, as shown in figure 3.6. To make the

method automatic, we propose stopping the Gauss-Newton iterations when

$$\frac{|\psi(\boldsymbol{x}_{k+1}) - \psi(\boldsymbol{x}_k)|}{\psi(\boldsymbol{x}_{k+1})} \leq \frac{\gamma}{2}.$$

We used as a good value for γ the level of noise in the observed data.



$\boldsymbol{y}$ error $\boldsymbol{x}$ error

Figure 3.5: Plots of the relative errors . Left is the error of the nonlinear parameter $\boldsymbol{y}$ during the Gauss-Newton iterations; right is the error of the image $\boldsymbol{x}$.

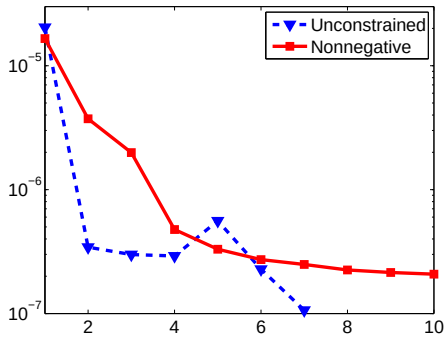


Figure 3.6: Objective function behavior

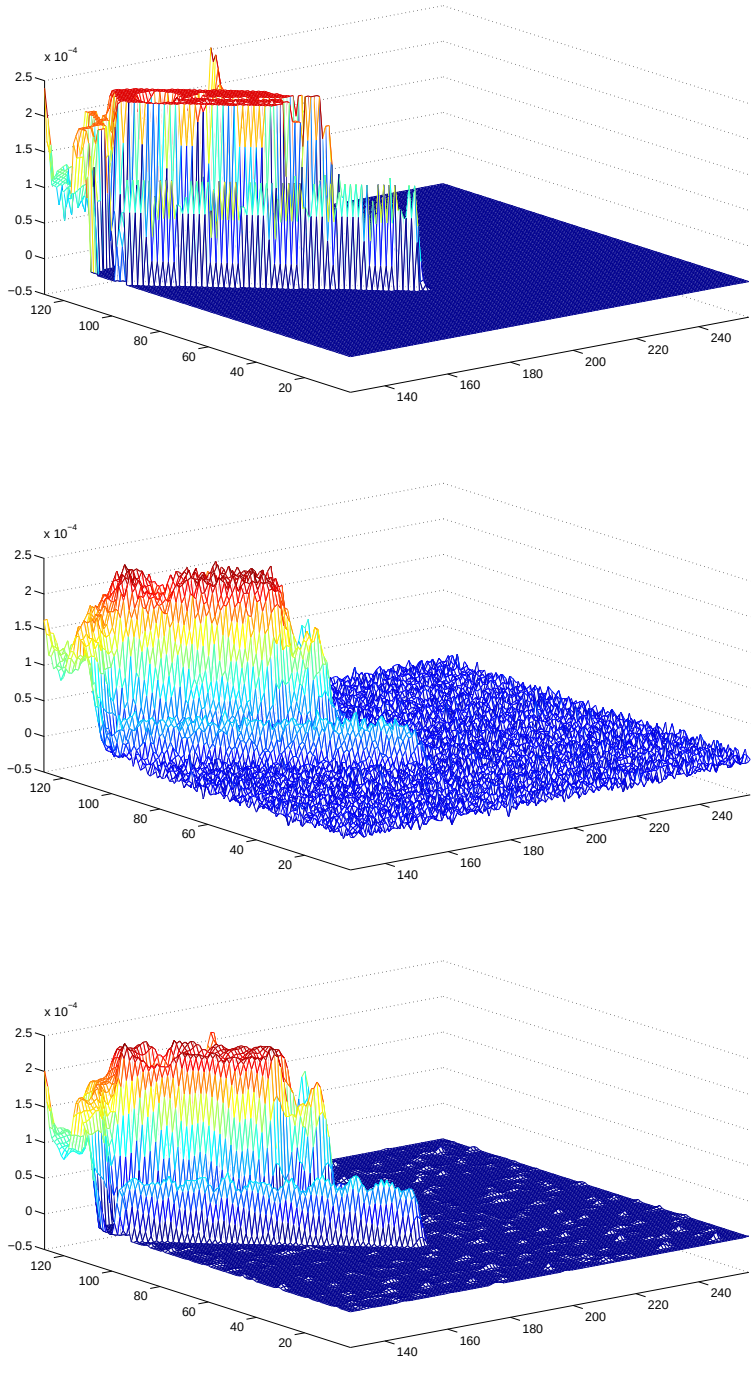


Figure 3.7: Surfaces of portions of the true object and the reconstructed images $\boldsymbol{x}$: Top is the satellite image. In the second row is the image solution of the unconstrained problem. Bottom is the image obtained by imposing nonnegative constraints.

3.1.2 Alternating Optimization

Also Alternating Optimization can be applied to solve the separable problem (3.6), modeling blind deconvolution. As previously said, the subproblem over $\boldsymbol{x}$ needs regularization. Unlike Variable Projection, Alternating Optimization allows both direct and iterative regularization, since the subproblem over $\boldsymbol{x}$ can be solved with any kind of accuracy. Clearly in order to save computational cost, an inexact solution, corresponding to iterative regularization, is preferable.

By iterative regularization we mean that the number of iterations plays the role of regularization parameter [29, 55]. Indeed iterative methods build up a sequence of vectors that first approaches the regularized solution but then converges to the naive solution of (2.3), which, as we already mentioned, is corrupted by noise. This behavior is known as *semiconvergence* and the way to obtain a meaningful solution consists of stopping the iterative procedure before the noise introduces artifacts and undesired effects on the reconstructions.

We repeat the experiment made in section 2.2 increasing the number of iterations to show the semiconvergent behavior in the plot of the relative error. In Figure 3.8 you can see as the curve reaches its minimum and then increases, both for the unconstrained and the constrained solver.

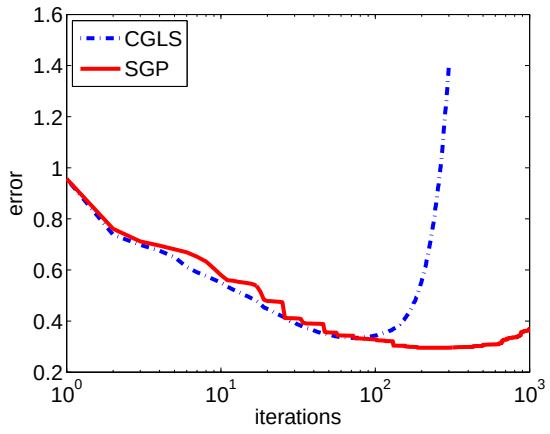


Figure 3.8: Semiconvergent behavior of iterative methods for the deconvolution problem $\min_{\boldsymbol{x}\geq\boldsymbol{0}} \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|^2$

A common rule to choose the stopping iteration k is the *discrepancy principle* [42]. It requires an estimation of the noise level $\delta = \|\boldsymbol{\eta}\|$; then the regularization parameter k should be chosen so that the norm of the residual vector is smaller or equal than the

noise level. By this we mean that one should choose the smallest k such that

$$\|\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}\| \leq \delta\,. \tag{3.18}$$

For large scale problems, like in imaging, iterative regularization can be favorable mostly because the structure of the coefficient matrix $\boldsymbol{A}$ is efficiently used for the matrix-vector products. Indeed for spatially invariant image deblurring, $\boldsymbol{A}$ is a block structured matrix; for instance, if periodic boundary conditions are assumed, $\boldsymbol{A}$ is a block circulant matrix with circulant blocks. The spectra decomposition of these special matrices makes matrix-vector multiplications really efficient, because they can be performed through FFT (Fast Fourier Transform) [31].

3.2 Numerical comparison between the two methods

In this section we perform both Variable Projection and Alternating Optimization to solve a blind deconvolution problem, stated as (3.6). Like in Chapter 1, in the scheme of Alternating Optimization we solve the subproblem over $\boldsymbol{y}$ through just one step of Gauss-Newton algorithm. We briefly recall that, especially in this case, the crucial difference between Variable Projection and Alternating Optimization concerns the computation of the Jacobian matrix. For a detailed comparison of the algorithms we refer to section 1.5; here we restrict ourselves to a summary of the algorithms, pointing out that the optimization problems, thus the objective functions, are different.

Furthermore, $\boldsymbol{x}_{\ell+1}$ will be computed through the proposed LBD-Regularized Reduced Projected-Newton method, described in section 2.3. We already mentioned that the algorithm does not move away from zero the components of the solution already in the active set. This is not a good feature when the aim is to well recover the image with its details; on the other hand, it can be suitable for blind deconvolution, where at each iteration ℓ a new blurring operator $\boldsymbol{A}(\boldsymbol{y}_\ell)$ is defined and consequently a new deconvolution problem must be solved.

Variable Projection to solve $\min_{\boldsymbol{y}} \psi(\boldsymbol{y}) = \|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x}(\boldsymbol{y}) - \boldsymbol{b}\|_2^2$	**Alternating Optimization** to solve $\min_{\boldsymbol{x}\geq 0,\boldsymbol{y}} f(\boldsymbol{x},\boldsymbol{y}) = \|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x} - \boldsymbol{b}\|_2^2$
given $\boldsymbol{y}_0$	given $\boldsymbol{y}_0$
for $\ell = 0, 1, 2, \ldots$	for $\ell = 0, 1, 2, \ldots$
1. $\boldsymbol{x}_{\ell+1} = \underset{\boldsymbol{x}\geq 0}{\operatorname{argmin}} \|\boldsymbol{A}(\boldsymbol{y}_\ell)\boldsymbol{x} - \boldsymbol{b}\|_2$	1. $\boldsymbol{x}_{\ell+1} = \underset{\boldsymbol{x}\geq 0}{\operatorname{argmin}} \|\boldsymbol{A}(\boldsymbol{y}_\ell)\boldsymbol{x} - \boldsymbol{b}\|_2$
2. $\boldsymbol{r}_\ell = \boldsymbol{A}(\boldsymbol{y}_\ell)\boldsymbol{x}_{\ell+1} - \boldsymbol{b}$	2. $\boldsymbol{r}_\ell = \boldsymbol{A}(\boldsymbol{y}_\ell)\boldsymbol{x}_{\ell+1} - \boldsymbol{b}$
3. $\boldsymbol{J}_{VP} = \boldsymbol{J}_{\boldsymbol{y}}(\psi(\boldsymbol{y})) = \boldsymbol{J}(f(\boldsymbol{x}(\boldsymbol{y}), \boldsymbol{y}))$	3. $\boldsymbol{J}_{AO} = \boldsymbol{J}_{\boldsymbol{y}}(f(\boldsymbol{x}, \boldsymbol{y}))$
4. $\boldsymbol{d}_\ell = \arg\min_{\boldsymbol{d}} \|\boldsymbol{J}_{VP}\boldsymbol{d} + \boldsymbol{r}_\ell\|_2$	4. $\boldsymbol{d}_\ell = \arg\min_{\boldsymbol{d}} \|\boldsymbol{J}_{AO}\boldsymbol{d} + \boldsymbol{r}_\ell\|_2$
5. determine τ_ℓ	5. determine τ_ℓ
6. $\boldsymbol{y}_{\ell+1} = \boldsymbol{y}_\ell + \tau_\ell \boldsymbol{d}_\ell$	6. $\boldsymbol{y}_{\ell+1} = \boldsymbol{y}_\ell + \tau_\ell \boldsymbol{d}_\ell$
end	end

In the whole section, we consider the Gaussian PSF, defined as

$$\mathcal{K}(s,t) = \frac{1}{2\pi\sqrt{\delta}} \exp\left(-\frac{1}{2} \begin{bmatrix} s & t \end{bmatrix} \begin{bmatrix} \sigma_1^2 & \rho^2 \\ \rho^2 & \sigma_2^2 \end{bmatrix}^{-1} \begin{bmatrix} s \\ t \end{bmatrix} \right) \tag{3.19}$$

where $\delta = \sigma_1^2\sigma_2^2 - \rho^4 > 0$. In this case the parameter vector $\boldsymbol{y}$ contains only three values, σ_1, σ_2 and ρ, describing respectively the width along the 2 directions and the orientation of the PSF. In particular we consider as a true objects the satellite and the grain images from the *Restore Tools*, blurred by the Gaussian PSF (3.19) defined by the parameter $\boldsymbol{y}_{true} = (\sigma_1, \sigma_2, \rho) = (1.5, 2, 0.5)$ and corrupted by 1% white Gaussian noise. The PSF is shown in figure 3.9.

Figure 3.10 contains the relative errors on the image $\boldsymbol{x}$ and the parameters $\boldsymbol{y}$ achieved by Variable Projection and Alternating Optimization for both satellite and grain test problems. In both the examples, Alternating Optimization seems to provide a smoother relative error behavior and to reach smaller error values, both for $\boldsymbol{x}$ and $\boldsymbol{y}$ reconstructions. We remark that both the Jacobian matrices $\boldsymbol{J}_{AO}$ and $\boldsymbol{J}_{VP}$ are computed through finite differences and that Variable Projection seems more sensitive than Alternating Optimization to the Jacobian approximation. The reconstructed images are shown in Figures 3.11 and 3.12.

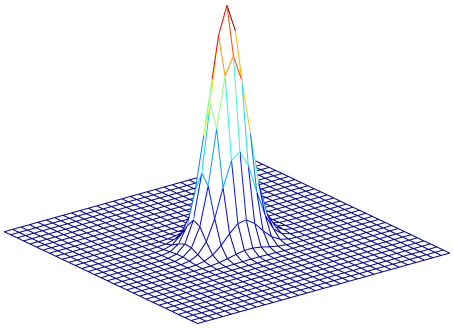


Figure 3.9: A Gaussian PSF as given in equation (3.19) with $\boldsymbol{y}_{\text{true}} = [\sigma_1, \sigma_2, \rho]^T$, where $\sigma_1 = 1.5$, $\sigma_2 = 2$, and $\rho = 0.5$.

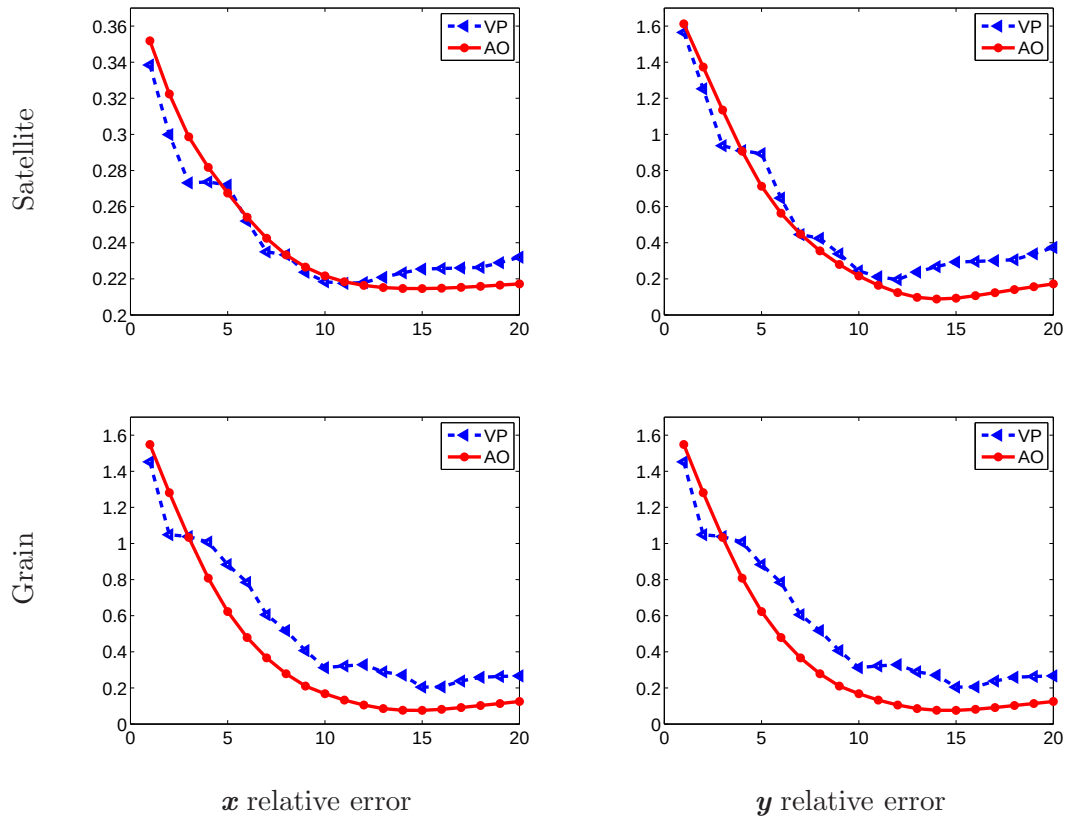


Figure 3.10: Relative error on $\boldsymbol{x}$ and $\boldsymbol{y}$ during the iterative procedure of Variable Projection (magenta dots) and Alternating Optimization (black triangles). First row is related to satellite image, second row to grain image.

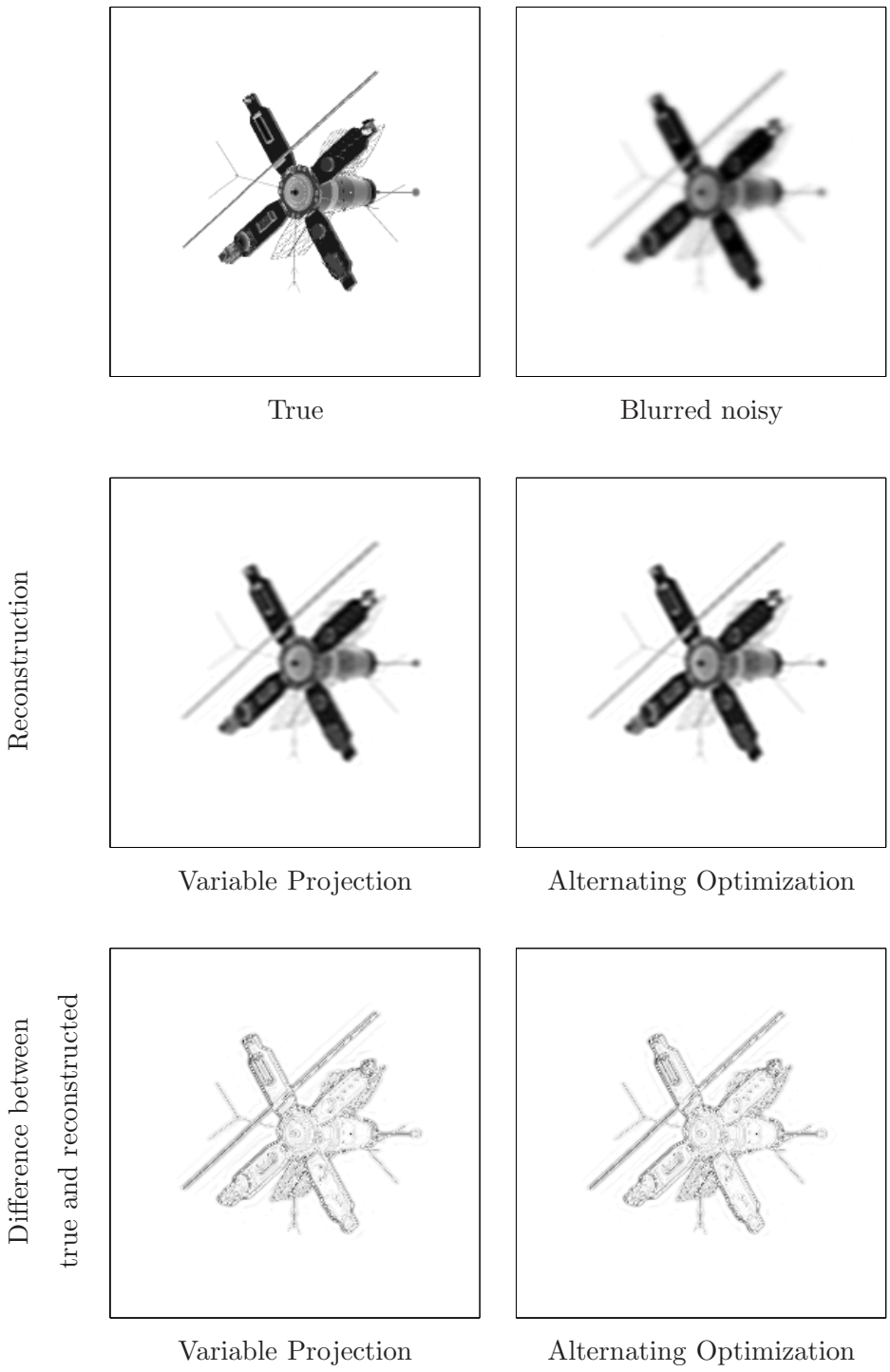


Figure 3.11: Results relative to satellite image. First row: true image and data. Second row: reconstructed image at iteration corresponding to the minimum error achieved by Variable Projection (on the left) and Alternating Optimization (on the right). Third row: difference between the true image and the reconstructed by Variable Projection (on the left) and Alternating Optimization (on the right).

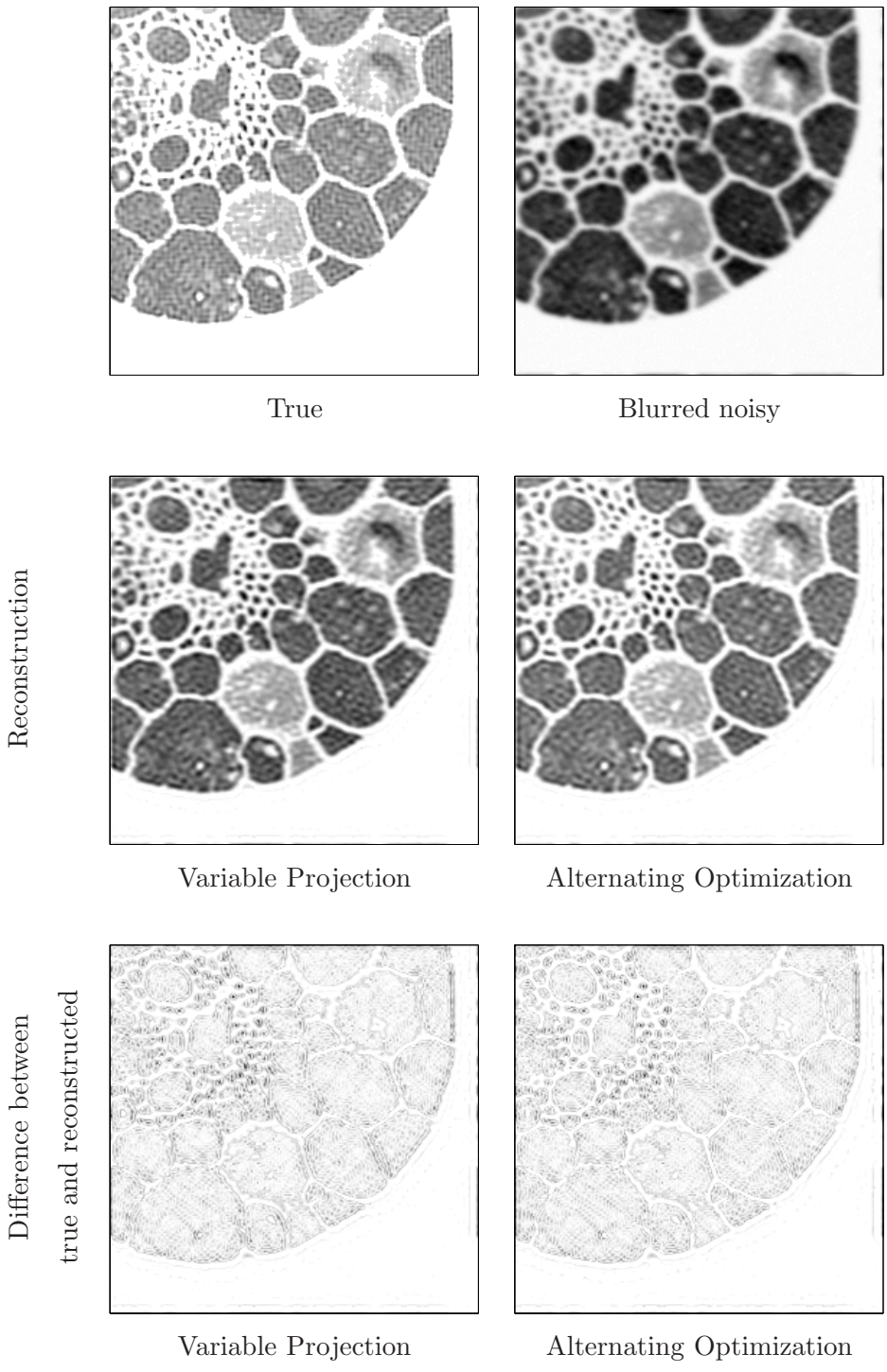


Figure 3.12: Results relative to grain image. First row: true image and data. Second row: reconstructed image at iteration corresponding to the minimum error achieved by Variable Projection (on the left) and Alternating Optimization (on the right). Third row: difference between the true image and the reconstructed by Variable Projection (on the left) and Alternating Optimization (on the right).

Chapter 4

Blind deconvolution for Fourier-based image restoration

In this chapter we address the image reconstruction problem from Fourier data with uncertainties on the spatial frequencies corresponding to the measured data. By considering such dependency on the frequencies as unknown, we obtain a blind deconvolution problem that can be formulated as a separable least squares problem.

An application to the framework of astronomical imaging of high energy radiation emitted during a solar flare is given; in particular, we address the hardware of the NASA RHESSI satellite, which provides some Fourier components of the unknown image at specific frequencies depending on the spacecraft rotation around its axis.

In a recent paper [12], Bonettini and Prato proposed a non-scaled gradient method as iterative regularization algorithm for solving the deconvolution problem when the PSF is given, showing significant improvements with respect to the other visibility-based image reconstruction methods developed for the RHESSI data analysis. In force of the results of the proposed method, in this work we use the same settings to solve the image reconstruction subproblem. Due to the performed iterative regularization, the Alternating Optimization approach results more useful for the RHESSI application than the Variable Projection.

4.1 Problem formulation

Image reconstruction in many fields, such as X-ray diffraction, electron microscopy, and diffraction optics, can be interpreted as the problem of estimating a function from its Fourier transform values.

Let us consider the n-vector image in the space domain as $\boldsymbol{x}$, evaluated at a set of grid points $(\boldsymbol{\xi}, \boldsymbol{\zeta})$, and the measured data $\boldsymbol{b} \in \mathbb{R}^m$, with the related available frequency sampling $(\boldsymbol{u}, \boldsymbol{v})$. The operator $\boldsymbol{A} : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ is linear and arises from the discretization of the Fourier transform, thus the k-th element of the vector resulting from its product with $\boldsymbol{x}$ is[1]

$$(\boldsymbol{A}\boldsymbol{x})_k = \sum_{j=1}^{n} x_j \exp\left(2\pi i(\xi_j u_k + \zeta_j v_k)\right), \quad k = 1, \ldots, m\,. \tag{4.1}$$

For some applications, the spatial frequencies $\boldsymbol{y} = (\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{R}^{2m}$ can be considered as parameters to be estimated. In this case we write the discrete Fourier transform as $\boldsymbol{A} = \boldsymbol{A}(\boldsymbol{y})$.

Fourier-based image restoration can be thus modeled as a separable inverse problem of the form

$$\boldsymbol{b} = \boldsymbol{A}(\boldsymbol{y})\boldsymbol{x} + \boldsymbol{\eta}\,, \tag{4.2}$$

where

- $\boldsymbol{b} \in \mathbb{C}^m$ contains the available complex data;
- $\boldsymbol{x} \in \mathbb{R}^n$ is the unknown true image. Since the pixels content in general refers to a physical quantity, we assume the entries of $\boldsymbol{x}$ to be real and non-negative;
- $\boldsymbol{y} \in \mathbb{R}^{2m}$ are the unknown spatial frequencies. It is realistic to assume a certain range of variability $[\boldsymbol{y}^{\min}, \boldsymbol{y}^{\max}]$ to be provided by the specific application;
- $\boldsymbol{A}$ is the discrete Fourier transform, which can be explicitly written as a $m \times n$ matrix

$$\boldsymbol{A}(\boldsymbol{y}) = \exp\left(2\pi i \begin{bmatrix} \xi_1 u_1 + \zeta_1 v_1 & \ldots & \xi_n u_1 + \zeta_n v_1 \\ \xi_1 u_2 + \zeta_1 v_2 & \ldots & \xi_n u_2 + \zeta_n v_2 \\ \vdots & & \vdots \\ \xi_1 u_m + \zeta_1 v_m & \ldots & \xi_n u_m + \zeta_n v_m \end{bmatrix}\right);$$

- $\boldsymbol{\eta}$ is a vector modeling additive noise on the measured data.

It follows from (4.2) that the problem of image and frequency reconstructions from Fourier data can be formulated as the following optimization problem

$$\min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y}) = \left\|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x} - \boldsymbol{b}\right\|^2_{\mathbb{C}^m} \quad \text{subject to} \quad \boldsymbol{x} \geq \boldsymbol{0}, \quad \boldsymbol{y}^{\min} \leq \boldsymbol{y} \leq \boldsymbol{y}^{\max}\,. \tag{4.3}$$

[1] Coherently with the definition given in [40, 47], we define the Fourier transform with a positive sign in the exponent.

Equation (4.3) is a separable least squares problem, which can be approached by the two methods we discussed in the previous chapters of this work.

The problem in (4.3) is a nonlinear, in general nonconvex, optimization problem, with $f(\cdot, \boldsymbol{y})$ convex for any $\boldsymbol{y}$. Moreover, the variables are naturally grouped in two separate convex sets and therefore the solution of (4.3) can be approached by the two methods we discussed in the previous chapters. We recall that Variable Projection requires the linear solution $\boldsymbol{x}$ to be exact; on the other hand, Alternating Optimization allows to compute inexactly both the $\boldsymbol{x}$ and $\boldsymbol{y}$ solutions. For this reason in section 4.2, when dealing with an application to Fourier-based blind deconvolution on simulated data from the RHESSI satellite, we prefer the Alternating Optimization approach, consisting of an iterative minimization with respect to each grouped variable:

$$\boldsymbol{x}_{\ell+1} = \underset{\boldsymbol{x} \geq \boldsymbol{0}}{\operatorname{argmin}} \; f(\boldsymbol{x}, \boldsymbol{y}_\ell) \tag{4.4}$$

$$\boldsymbol{y}_{\ell+1} = \underset{\boldsymbol{y}^{\min} \leq \boldsymbol{y} \leq \boldsymbol{y}^{\max}}{\operatorname{argmin}} \; f(\boldsymbol{x}_{\ell+1}, \boldsymbol{y}) \tag{4.5}$$

In order to avoid the expensive computation of an exact minimum point of $f(\boldsymbol{x}_\ell, \cdot)$ and $f(\cdot, \boldsymbol{y}_\ell)$, a common strategy in the blind deconvolution framework [19, 23] consists of the application of an iterative method to both subproblems (4.4) and (4.5), stopping the iterations when some criterion is satisfied.

Driven by the proposal given in [33], where the subproblem over $\boldsymbol{y}$ is solved through just one step of Newton's method, in order to solve (4.5) we perform one step of the Gauss-Newton algorithm, motivating this choice with the smaller computational costs. Indeed the Gauss-Newton requires the computation of the Jacobian matrix, thus avoiding the 2nd-order derivatives.
The Jacobian related to the residual of f is $\boldsymbol{J} = \dfrac{\partial \boldsymbol{A}(\boldsymbol{y})}{\partial \boldsymbol{y}} \boldsymbol{x}$; in particular, since $\dfrac{\partial \boldsymbol{A}(\boldsymbol{y})}{\partial y_k}$ is a $2m \times n$ matrix with null rows except for the k-th one, the Jacobian results to be a diagonal matrix, whose non-zero entries given by

$$\boldsymbol{J}_{kk} = \begin{cases} \sum_{j=1}^{n} 2\pi i \xi_j \exp(2\pi i(\xi_j u_k + \zeta_j v_k) x_j & \text{if } 1 \leq k \leq m \\ \sum_{j=1}^{n} 2\pi i \zeta_j \exp(2\pi i(\xi_j u_k + \zeta_j v_k) x_j & \text{if } m+1 \leq k \leq 2m \end{cases}.$$

In the next section we describe the hardware of the NASA RHESSI satellite and then provide some numerical results on simulated data.

4.2 Imaging with RHESSI

The solar satellite RHESSI [21] has been launched by NASA on February 5 2002 with the aim of providing new insights for the comprehension of the acceleration mechanisms

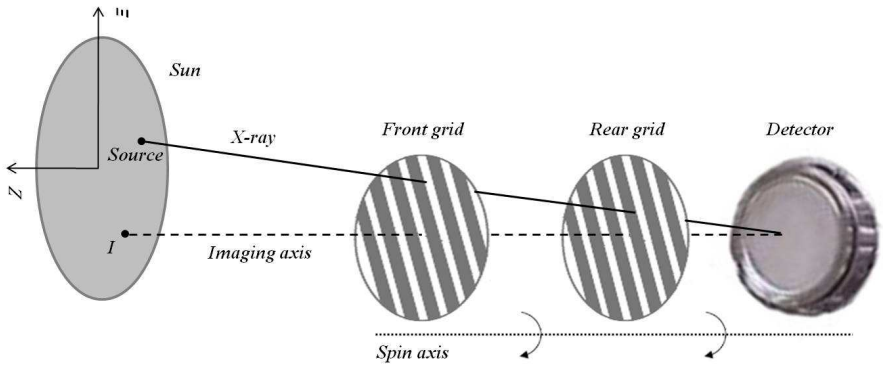


Figure 4.1: Schematic representation of a rotating modulation collimator.

occurring during solar flares [22]. Thanks to the spacecraft imaging system, RHESSI may provide both X-ray two-dimensional images with an angular resolution from 2 to 7 arcseconds and X-ray one-dimensional spectra with a spectral resolution from 0.5 to 2 keV. For the first time, with this mission an imaging-spectroscopy analysis of solar data is possible, whereby high resolution spectroscopy at each point of the X-ray image will allow spectral changes to be measured as the electrons propagate along the magnetic field in the flaring loop. Instead of the typical approaches for focusing optics wavelengths, which are impractical when dealing with X-rays, RHESSI exploits a bi-grid collimator strategy, which allows a partial transmission of the incident photons depending on their incident direction. More precisely, RHESSI observes X-ray emission from the Sun through a set of nine co-aligned pairs of rotating modulation collimators (RMCs) [20], and the transmitted radiation is recorded on a set of cooled HPGe detectors (see Figure 4.1). The raw data provided by RHESSI are, therefore, nine count profiles of the detected radiation as a function of time, modulated by the grids pairs. Several reconstruction algorithms have been developed since RHESSI's launch for recovering the X-ray image from the count profiles, such as CLEAN and Pixon [20]. However, a growing interest has been devoted recently to an alternative approach, which consists of a) exploiting the combination between rotation and modulation in order to estimate some image Fourier components at specific spatial frequencies, and b) using Fourier-based inversion algorithms to restore the image [12, 13, 40]. In the following we summarize the ideas at the basis of this approach and we propose an innovative strategy for the image recovery.

4.2.1 From counts to visibilities

In this section we want to give an idea about how the modulated count profiles measured by RHESSI can lead to an estimate of the unknown image's Fourier transform values, also

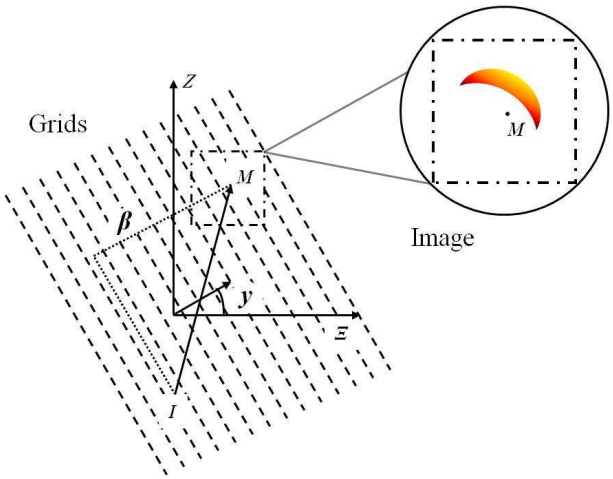


Figure 4.2: RHESSI imaging geometry. At each moment, the roll angle y and the aspect phase β are defined according to the imaging axis and the grids position.

called *visibilities* [41], in a set of spatial frequencies. A more detailed review about the concept of visibility within the RHESSI mission and the related available reconstruction algorithms is beyond the purpose of this thesis and has being carried out in parallel.
We define a reference system (Ξ, Z) on the Sun and we consider a flux distribution $x(\xi, \zeta)$, where (ξ, ζ) belongs to the image domain $\mathcal{G}$. The center of the image will be denoted by $M = (\xi_M, \zeta_M)$. Moreover, we restrict our analysis to a single collimator and we denote by $I(t) = (\xi_I(t), \zeta_I(t))$ the known time-dependent projection onto the solar disk of a reference point fixed with respect to the grids (also called *imaging axis*). With these settings, at time t the following two quantities are defined (see Figure 4.2):

- the *roll angle*, denoted by $y(t)$, which is the angle between the Ξ axis and the direction orthogonal to the grids orientation. Such angle is independent on the pixels of the image and represents the rotation of the spacecraft around its axis;
- the *aspect phase*, denoted by $\beta(t)$, which is the component orthogonal to the grids of the vector linking $I(t)$ with the map center M. Such angle depends on the grid's pitch p (i.e., the width of the couple slit+slat) and can be expressed as

$$\beta(t) = \frac{2\pi}{p}\{(\xi_I(t) - \xi_M)\cos(y(t)) + (\zeta_I(t) - \zeta_M)\sin(y(t))\}. \tag{4.6}$$

Following [20], the modulation of the incident radiation leads to a measured count profile in the form

$$C(t) = K \int_{\mathcal{G}} x(\xi, \zeta)(a_0 + a_1 \cos(\phi(\xi, \zeta, t)))d\xi d\zeta, \tag{4.7}$$

where K, a_0, a_1 are known hardware-dependent constants and

$$\phi(\xi,\zeta,t) = \frac{2\pi}{p}\{(\xi - \xi_I(t))\cos(y(t)) + (\zeta - \zeta_I(t))\sin(y(t))\}. \tag{4.8}$$

If we introduce the visibility function

$$V(u,v) = \int_{\mathcal{G}} x(\xi,\zeta)\exp\left(2\pi i(u(\xi-\xi_M) + v(\zeta-\zeta_M))\right)d\xi d\zeta, \qquad (u,v)\in\mathbb{R}^2, \tag{4.9}$$

and we consider the split

$$\phi(\xi,\zeta,t) = \phi'(\xi,\zeta,t) - \beta(t), \tag{4.10}$$

with

$$\phi'(\xi,\zeta,t) = \frac{2\pi}{p}\{(\xi - \xi_M)\cos(y(t)) + (\zeta - \zeta_M)\sin(y(t))\}, \tag{4.11}$$

then from equations (4.7) and (4.9) it follows that the relation

$$C(t) = Ka_0\int_{\mathcal{G}} x(\xi,\zeta)d\xi d\zeta + Ka_1\Re(y(t))\cos(\beta(t)) + Ka_1\Im(y(t))\sin(\beta(t)) \tag{4.12}$$

holds, where $\Re(y)$, $\Im(y)$ are the real and imaginary part of $V(u_y, v_y)$ and

$$(u_y, v_y) = \left(\frac{\cos(y)}{p}, \frac{\sin(y)}{p}\right). \tag{4.13}$$

4.2.2 The data stacking

In the previous section we introduced the continuous relation linking the count profiles with the visibilities at some specific spatial frequencies. Since in practice we have to deal with discrete quantities, the following procedure to translate counts into visibilities is adopted (see Figure 4.3):

- each count bin detected by RHESSI is labeled with the corresponding roll angle and aspect phase. For a given time range, a plot of the counts as functions of y and β is generated;
- the roll angle and aspect phase values are discretized into two vectors

  $$(\overline{y}_0,\ldots,\overline{y}_m) \quad , \quad (\overline{\beta}_0,\ldots,\overline{\beta}_\ell)$$

  and a matrix is created by summing up all the counts belonging to the same cell generated by the discretization;

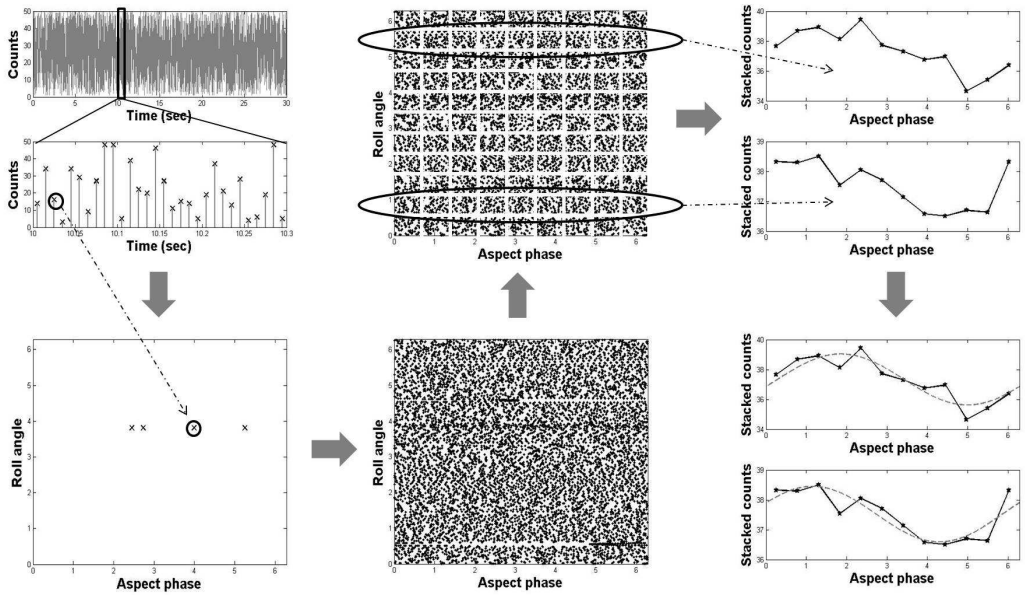


Figure 4.3: Schematic representation of the data stacking process: a) counts are discretized into time bins; b) a roll angle y versus aspect phase β map is populated; c) the y and β ranges are discretized in bins; d) for each roll bin, the stacked counts as a function of β are fitted with equation (4.14).

- for each roll bin $[\overline{y}_{k-1}, \overline{y}_k]$ $(k = 1, \ldots, m)$, the corresponding row of the matrix is fitted with the function

$$F + R\cos(\boldsymbol{\beta}) + I\sin(\boldsymbol{\beta}), \tag{4.14}$$

  with $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_\ell)$, $\beta_i \in [\overline{\beta}_{i-1}, \overline{\beta}_i]$ $(i = 1, \ldots, \ell)$, and, since the parameters K, a_1 are known, from R, I and by means of equation (4.12) the value of $V(u_{y_k}, v_{y_k})$ follows, where $y_k \in [\overline{y}_{k-1}, \overline{y}_k]$ $(k = 1, \ldots, m)$.

The key point of our work is that the actual value of the roll angle y_k is unknown, since only the existence of such a value in the interval $[\overline{y}_{k-1}, \overline{y}_k]$ is theoretically ensured (essentially for the First Mean Value Theorem for Integration). What is typically done in the existing reconstruction algorithms is to associate arbitrarily the computed visibility to the frequencies (u_{y_k}, v_{y_k}) defined by equation (4.13), where y_k is the middle point of $[\overline{y}_{k-1}, \overline{y}_k]$.
In the following we will remove such assumption, and the value of $y_k \in [\overline{y}_{k-1}, \overline{y}_k]$ will be treated as a further unknown of the reconstruction problem. Figure 4.4 represents a typical example of frequencies samples.

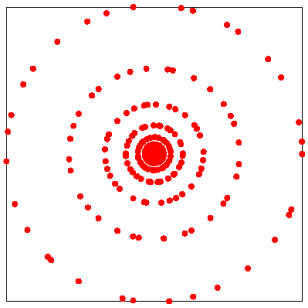


Figure 4.4: Example of a typical sampling of RHESSI data in the frequency plane

4.2.3 Formulation of the RHESSI blind deconvolution problem

Following the remarks observed in the previous section, we want to reformulate the RHESSI image restoration problem from visibilities proposed in [12] and resulting in the minimization problem

$$\min_{\boldsymbol{x}\geq 0} f(\boldsymbol{x}) = \left\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\right\|^2_{\mathbb{C}^m}, \tag{4.15}$$

where:

1. the vector $\boldsymbol{x}$ represents the image of the emitted flux, corresponding to the flux distribution $\boldsymbol{x}(\boldsymbol{\xi}, \boldsymbol{\zeta})$ evaluated at a set of grid points (ξ_j, ζ_j) $(j = 1, \ldots, n)$. Since the pixels content refers to a physical quantity, the entries of $\boldsymbol{x}$ have to be real and non-negative;

2. the vector $\boldsymbol{b}$ contains the computed complex visibilities provided by the stacking and fitting process described in the previous section, for all the detectors (thus, if m_i is the number of roll bins selected for the i-th collimator, then $m = m_1 + \ldots + m_9$);

3. the linear operator $\boldsymbol{A}$ arises from the discretization of the Fourier transform and, for a given $\boldsymbol{x} \in \mathbb{R}^n$, is defined as

$$(\boldsymbol{A}\boldsymbol{x})_k = \sum_{j=1}^{n} x_j \exp\left(2\pi i(\xi_j u_k + \zeta_j v_k)\right), \quad k = 1, \ldots, m, \tag{4.16}$$

with $(u_k, v_k) = \left(\frac{1}{p_k}\cos\left(\frac{\overline{y}_{k-1} + \overline{y}_k}{2}\right), \frac{1}{p_k}\sin\left(\frac{\overline{y}_{k-1} + \overline{y}_k}{2}\right)\right)$. Here p_k is the pitch of the detector corresponding to the frequency (u_k, v_k).

If we take into account of the unknown dependency of the spatial frequencies on the roll angle vector $\boldsymbol{y} = (y_1, \ldots, y_m)$, and we make it explicit in the problem formulation,

then equation (4.15) assumes the form

$$\begin{aligned} &\min && f(\boldsymbol{x},\boldsymbol{y}) = \left\|\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x}-\boldsymbol{b}\right\|^2_{\mathbb{C}^m}, \\ &\boldsymbol{x}\geq \boldsymbol{0} && \\ &\boldsymbol{y}^{\min}\leq \boldsymbol{y}\leq \boldsymbol{y}^{\max} && \end{aligned} \tag{4.17}$$

where $\boldsymbol{y}^{\min} = (\overline{y}_0,\ldots,\overline{y}_{m-1})$, $\boldsymbol{y}^{\max} = (\overline{y}_1,\ldots,\overline{y}_m)$, and the linear operator $\boldsymbol{A}(\boldsymbol{y})$ is now defined by the relation

$$(\boldsymbol{A}(\boldsymbol{y})\boldsymbol{x})_k = \sum_{j=1}^{n} \boldsymbol{x}_j \exp\left(\frac{2\pi i}{p_k}(\xi_j\cos(y_k)+\zeta_j\sin(y_k))\right), \quad k=1,\ldots,m. \tag{4.18}$$

As we already mentioned in chapter 1, the presence of constraints on the nonlinear variable $\boldsymbol{y}$ does not affect the Alternating Optimization method, except for the employment of an appropriate projection method for the solution of the subproblem over $\boldsymbol{y}$.
The Jacobian matrix $\boldsymbol{J}$ of the residual of f in (4.17) is diagonal, with diagonal elements equal to

$$\boldsymbol{J}_{kk} = \sum_{j=1}^{n} \frac{2\pi i}{p_k}\left(-\sin(\alpha_k)\,\xi_j+\cos(\alpha_k)\,\zeta_j\right)\exp\left(\frac{2\pi i}{p_k}\left(\xi_j\cos(\alpha_k)+\zeta_j\sin(\alpha_k)\right)\right).$$

4.3 Numerical experiments

In this section we want to investigate the effectiveness of the blind deconvolution approach in reconstructing X-ray images of solar flares from RHESSI data. The evaluation of the results will be carried out in comparison with the Space-D algorithm proposed in [12], which solves the single deconvolution problem where $\boldsymbol{y}$ is set equal to the middle point of the roll bins. To this aim, we built up three simulated datasets by following a strategy similar to the one proposed in [12]. In particular,

- we considered a real flare event (23 July 2002, 00:29:10–00:30:19 UT) and, starting from the radiation collected by RHESSI in three different energy ranges, we reconstructed the corresponding images with the CLEAN algorithm [20], available within the solar software (SSW) of the mission. Such images, suitably cleaned up by possible artifacts introduced by the reconstruction method, have been considered as our target distributions;

- we considered the detectors from 3 to 9, which is a usual choice when dealing with RHESSI images (see e.g. [46, 47, 56]), and for the k-th subcollimator ($k=3,\ldots,9$) we simulated a typical uniform discretization $(\overline{y}_{0,k},\ldots,\overline{y}_{m_k,k})$ of the roll angle

values. In particular, $(m_3, \ldots, m_9)$ has been set equal to $(32, 32, 32, 30, 22, 12, 6)$, thus leading to a total number of $m = 166$ visibilities;

- in order to mimic the RHESSI data stacking process, for each roll bin $[\overline{y}_{i-1,k}, \overline{y}_{i,k}]$ $(i = 1, \ldots, m_k)$, we chose a random roll angle $y_{i,k}$ and we calculated the corresponding visibilities through numerical integration of the Fourier transform by means of the SSW routine *hsi_vis_map2vis.pro*. We will denote the resulting three synthetic datasets as Sim1, Sim2 and Sim3. The visibilities in such datasets will be characterized by the presence of a systematic source of noise only, due to the fact that in the inversion procedure the exact values of $y_{i,k}$ are unknown;

- finally, we corrupted such visibilities by realistic statistical noise through the SSW routine *hsi_vis_randomize.pro*. These latter datasets will be denoted by Sim1_N, Sim2_N and Sim3_N.

We first consider the Sim1, Sim2 and Sim3 cases, in which no statistical noise is present on the simulated visibilities. Besides all the parameters described in the previous section, the Alternating Optimization scheme is initialized with a vector $\boldsymbol{y}^{(0)}$ equal to the middle points of each roll bin. In these settings, the image reconstructed by the Space-D algorithm is exactly $\boldsymbol{x}^{(1)}$.
In Figures 4.5, and 4.8, we report the target images, and frequencies respectively, (first row) and the reconstructions obtained with Space-D (second row) and the blind deconvolution formulation (third row). Moreover, in Figure 4.6 we show the relative reconstruction errors in Euclidean norm of the image (first row) and the roll angle array (second row), as functions of the cycle number. From the results obtained we can see that the blind deconvolution approach is able to improve the image quality with respect to the Space-D algorithm in all the considered datasets, thanks to a better estimate of the underlying roll angle array $\boldsymbol{y}$. Although the restored images appear to be very similar, a better separation of the two strongest sources in Sim2 can be noticed, together with a higher resolution in the two Northern compact sources in Sim3 (see Figure 4.5). Further considerations can be done by analyzing the results obtained in presence of statistical noise, which are reported in Figures 4.7, 4.9 and 4.10. In particular, we can see that:

- for the Sim1_N and Sim2_N datasets, the blind deconvolution formulation still leads to some improvements in the final image, attested by a lower reconstruction error with respect to an approach with a fixed roll angle array;

- the blind deconvolution approach does not lead to significant improvements in the Sim3_N simulation. We point out that, in this case, the target image is the one

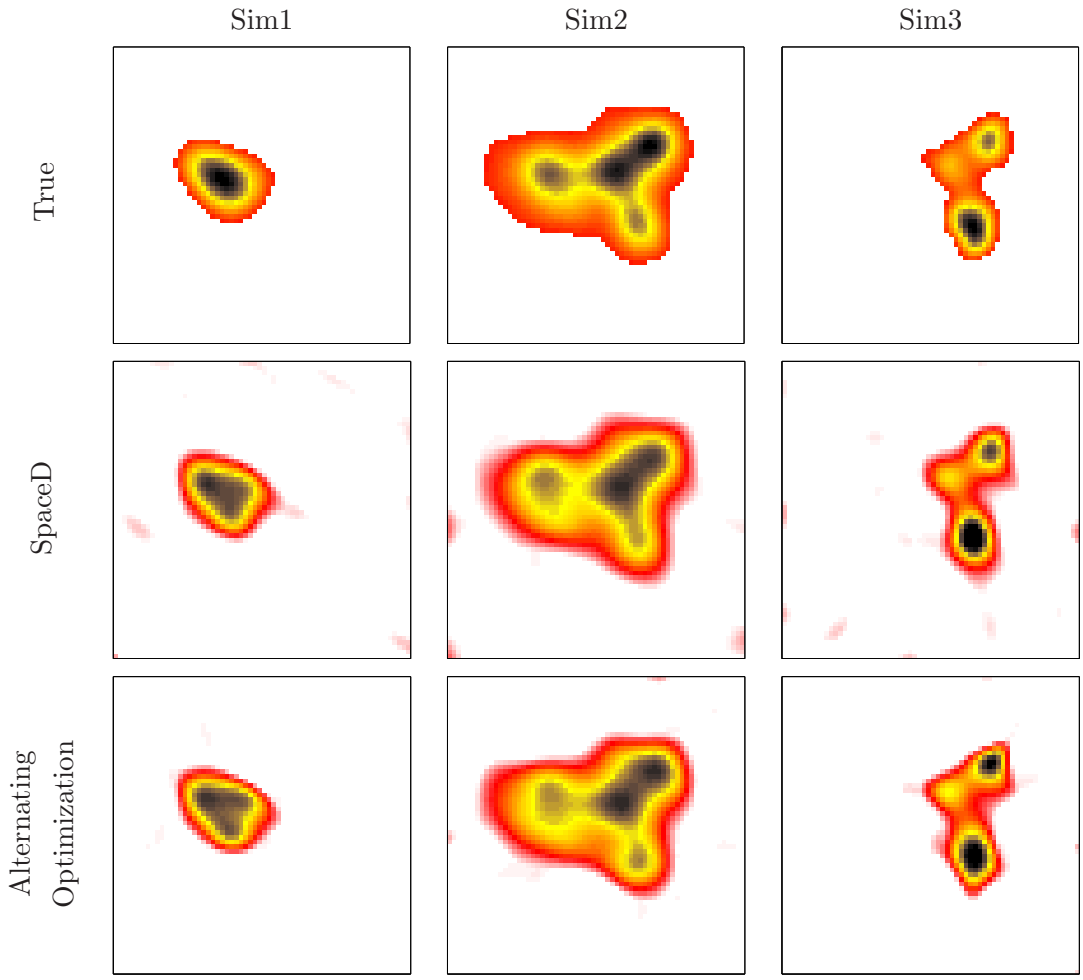


Figure 4.5: Results of the three simulated tests considered in the paper with noise-free data: Sim1 (first column), Sim2 (second column) and Sim3 (third column). From the first to the last row, the theoretical image and the reconstructions with Space-D and Alternating Optimization (AO) are presented, respectively.

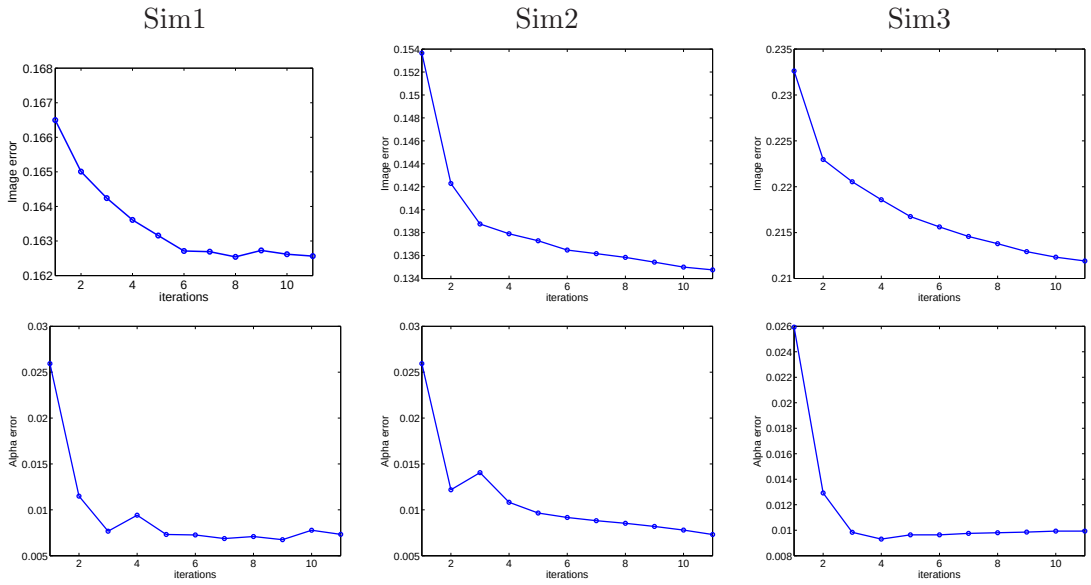


Figure 4.6: Relative reconstruction errors on the image (top row) and $\boldsymbol{y}$ (bottom row) as a function of the number of cycles for the Alternating Optimization method, in the case of noise-free data. The left (resp., central, right) column refers to the Sim1 (resp., Sim2, Sim3) dataset. The analogous values for the Space-D algorithm correspond to the first point of each plot.

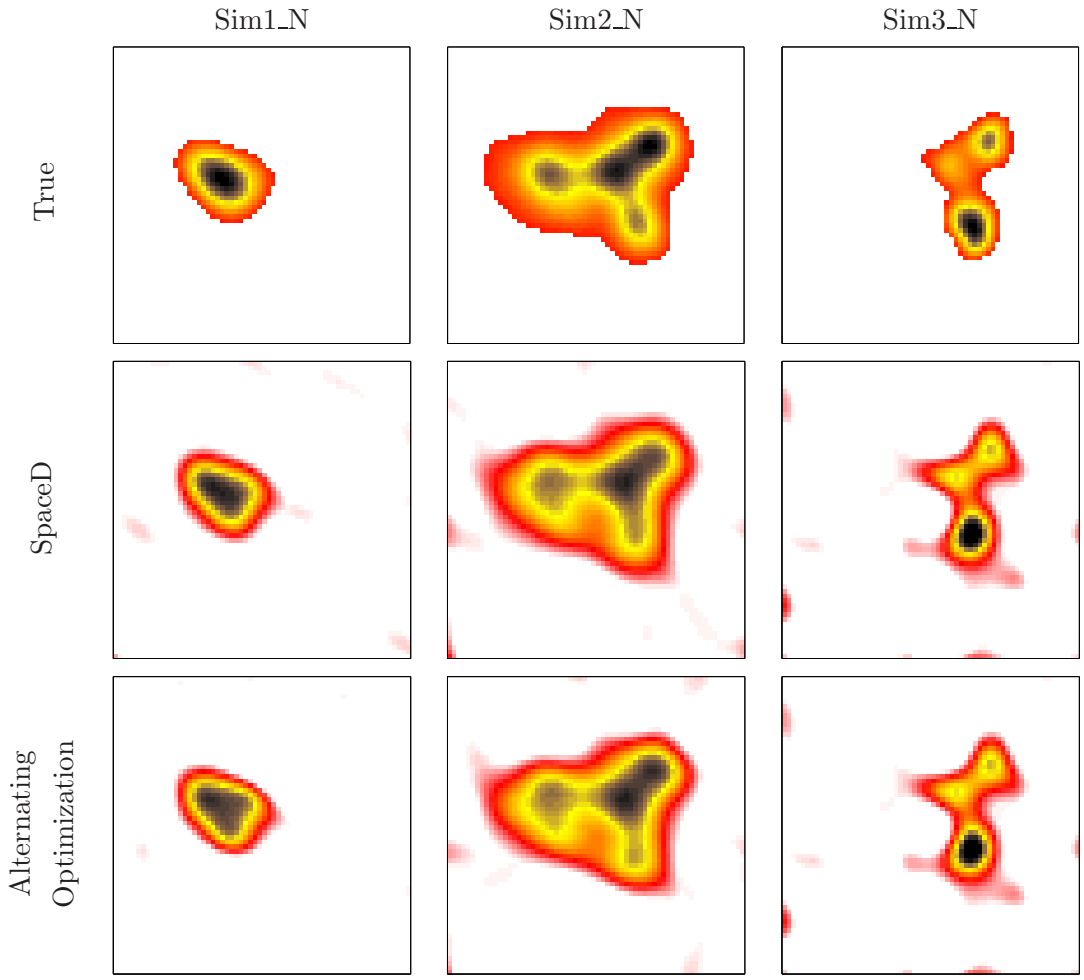


Figure 4.7: Results of the three simulated tests considered in the paper with noisy data: Sim1_N (first column), Sim2_N (second column) and Sim3_N (third column). From the first to the last row, the theoretical image and the reconstructions with Space-D and Alternating Optimization (AO) are presented, respectively.

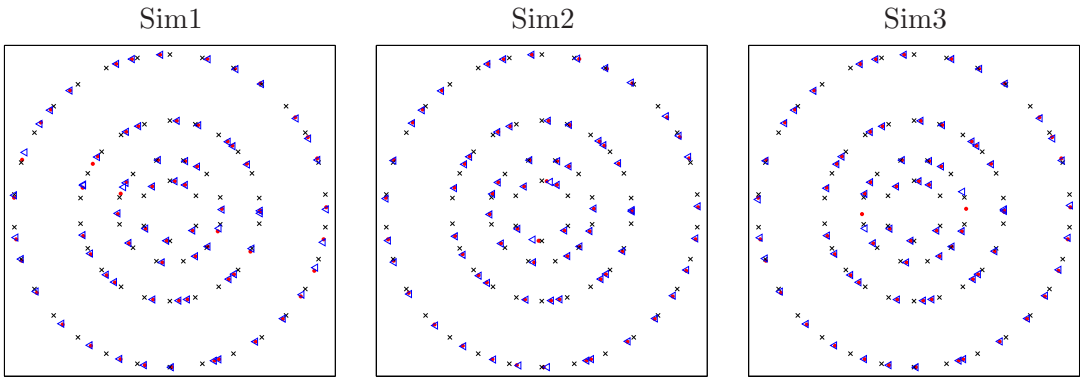


Figure 4.8: Frequencies reconstructions in the noise free case: plot of $\boldsymbol{u}$ versus $\boldsymbol{v}$ (4 of the 9 circles). Red points are the true values; black crosses are the values fixed as for SpaceD; blue triangles are the reconstructed values.

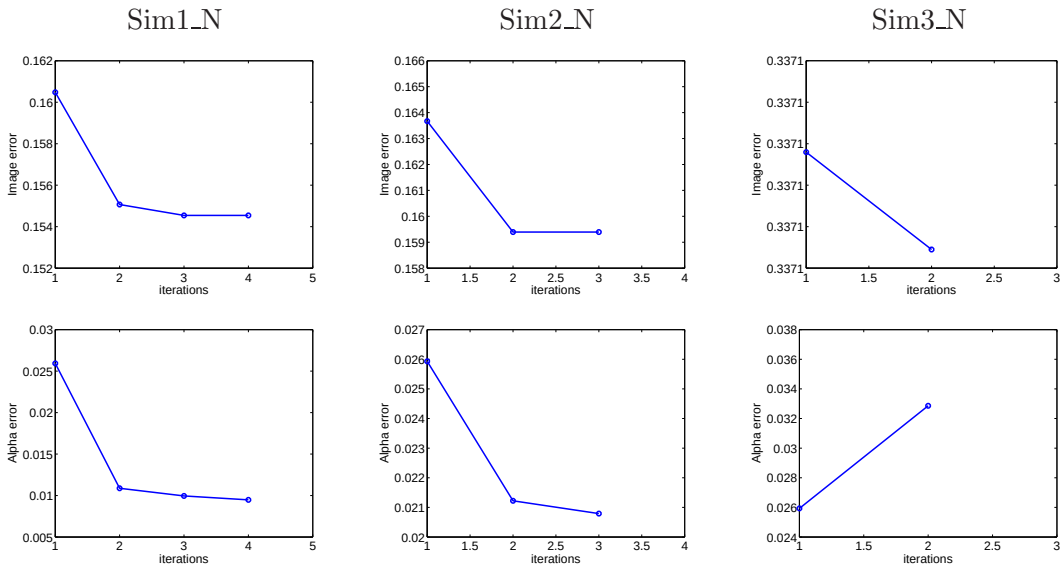


Figure 4.9: Relative reconstruction errors on the image $\boldsymbol{x}$ (top row) and the frequencies $\boldsymbol{y}$ (bottom row) as a function of the number of cycles for Alternating Optimization method, in the case of noisy data. The left (resp., central, right) column refers to the Sim1_N (resp., Sim2_N, Sim3_N) dataset. The analogous values for the Space-D algorithm correspond to the first point of each plot.

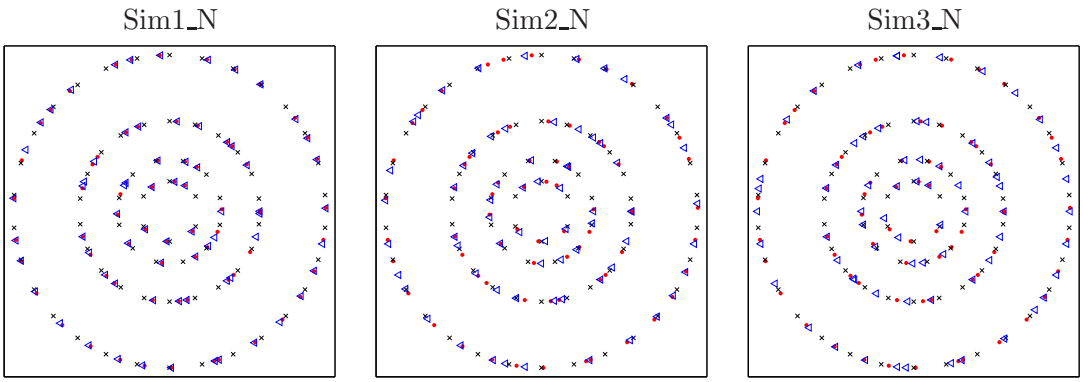


Figure 4.10: Frequencies reconstructions for noisy data: plot of $\boldsymbol{u}$ versus $\boldsymbol{v}$ (4 of the 9 circles). Red points are the true values; black crosses are the values fixed as for SpaceD; blue triangles are the reconstructed values.

with the lower radiation. Therefore, the corruption of the Sim3_N input visibilities introduced by the Poissonian-like statistical noise overcomes the systematic error related to a bad choice for $\boldsymbol{y}$. As a result of this, the regularizing effect achieved by Space-D algorithm provides already the best reconstruction. It is worth to point out that our semi-blind approach seems to "recognize" such situations, and automatically arrests the iterations at the second cycle.

Tables 4.1 and 4.2 resume all the relative errors on the image and the frequencies.

	Sim1	Sim2	Sim3	Sim1_N	Sim2_N	Sim3_N
SpaceD	0.166498	0.153648	0.232621	0.160479	0.163671	0.337099
Alternating Optimization	0.162563	0.133225	0.211914	0.154544	0.159393	0.337087

Table 4.1: Relative errors on the reconstructed images $\boldsymbol{x}$

	Sim1	Sim2	Sim3	Sim1_N	Sim2_N	Sim3_N
SpaceD	0.025931	0.025931	0.025931	0.025931	0.025931	0.025931
Alternating Optimization	0.007331	0.005895	0.009926	0.009483	0.020791	0.032859

Table 4.2: Relative errors on the reconstructed frequencies $\boldsymbol{y}$

Conclusions

In this thesis we discuss separable nonlinear least squares problems and give both a theoretical and numerical comparison of the two approaches commonly used in literature to address them. An application in the imaging framework is considered, that is the blind deconvolution problem. Our interest concerns constrained separable least squares; in particular, due to the features of the imaging application we deal with, we focus our attention on the case of nonnegative constraints imposed on a subset of variables.

The Variable Projection and Alternating Optimization methods are described, giving special attention to a constrained formulation, handled by both of these approaches. An original extension of the Variable Projection method to the special case of nonnegative constraints and a new formula for the computation of the Jacobian matrix are proposed. Furthermore, the main differences between the two methods are analyzed in details and, in order to concur to this aim, a simple numerical example is given.

Moreover we introduce to the image deblurring framework, pointing out the problem ill-posedness and the consequent need of regularization. To this aim, a regularized optimization algorithm for linear least squares problems is proposed; two different versions, for small and large scale examples, are provided. Due to the nature of the algorithm, we point out that it is particularly suited for blind deconvolution applications.

Furthermore, we state blind deconvolution as a separable problem and approach it through Variable Projection and Alternating Optimization. The numerical results show the significant improvement achieved by the nonnegative blind deconvolution formulation; moreover, Alternating Optimization seems to provide smaller errors than Variable Projection in the variables reconstructions.

From the numerical experiments conducted we can conclude that, when the exact Jacobian is available, Variable Projection performs better than Alternating Optimization; but, due to the fact that Variable Projection seems more sensitive to the Jacobian approximation, for application allowing the computation of the Jacobian just through finite differences, Alternating Optimization results preferable.

Finally, we address the blind deconvolution problem deriving from Fourier data with

uncertainties on the spatial frequencies corresponding to the measured data and give an application to the framework of astronomical imaging of high energy radiation emitted during a solar flare. Numerical tests on simulated data show that the blind deconvolution approach, corresponding to unknown spatial frequencies, provides some improvements in the reconstruction with respect to the formulation with fixed frequencies.

Bibliography

[1] J. Bardsley and C.R. Vogel. A nonnegativly constrained convex programming method for image reconstruction. *SISC*, 25(4):1326–1343, 2004.

[2] J. M. Bardsley and J. G. Nagy. Covariance-preconditioned iterative methods for nonnegatively constrained astronomical imaging. *simax*, 27:1184–1197, 2006.

[3] J.M. Bardsley and J. Goldes. Regularization parameter selection methods for ill-posed Poisson maximum likelihood estimation. *Inverse Problems*, 25(9), 2009.

[4] J.M. Bardsley, J. Merikoski, and R. Vio. The Stabilizing Properties of Nonnegativity Constraints in Least-Squares Image Reconstruction. *IJPAM*, 43(1):95–109, 2008.

[5] F. Benvenuto, R. Zanella, L. Zanni, and M. Bertero. Nonnegative least-squares image deblurring: improved gradient projection approaches. *Inverse Problems*, 26:025004, 2010.

[6] M. Bertero and P. Boccacci. *Introduction to inverse problems in imaging*. IoP, 1998.

[7] D. P. Bertsekas. Projected newton methods for optimization problems with simple constraints. *SIAM J. Control and Optimization*, 20:221–246, 1982.

[8] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, 1999.

[9] J. C. Bezdek, R. J. Hathaway, R. E. Howard, C. A. Wilson, and M. P. Windham. Local convergence analysis of a grouped variable version of coordinate descent. *Journal of Optimization Theory and Applications*, 54:471–477, 1987. 10.1007/BF00940196.

[10] J.C. Bezdek and R.J. Hathaway. *Some Notes on Alternating Optimization*, volume 2275 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2002.

[11] S. Bonettini. Inexact block coordinate descent methods with application to non-negative matrix factorization. *IMA Journal of Numerical Analysis*, 31(4):1431–1452, 2011.

[12] S. Bonettini and M. Prato. Nonnegative image reconstruction from sparse Fourier data: a new deconvolution algorithm. *Inverse Problems*, 26:095001, 2010.

[13] S. C. Bong, J. Lee, D. E. Gary, and H. S. Yun. Spatio-spectral maximum entropy method: I. Formulation and test. *Astrophys. J.*, 636(2):1159, 2006.

[14] T. F. Chan and Wong C. K. Total variation blind deconvolution. *IEEE Transactions on image processing*, 7:370–375, 1998.

[15] T. F. Chan and C. K. Wong. Convergence Of The Alternating Minimization algorithm for Blind Deconvolution. *Linear Algebra Appl*, 316:259–285, 2000.

[16] J. Chung, E. Haber, and J. Nagy. Numerical methods for coupled super-resolution. *Inverse Problems*, 22(4):1261–1272, 2006.

[17] J. Chung, J.G. Nagy, and D.P. O'Leary. A Weighted GCV Method for Lanczos Hybrid Regularization. *Elec. Trans. Num. Anal.*, 28:149–167, 2008.

[18] J. Chung, P. Sternberg, and C. Yang. High performance 3-D image reconstruction for molecular structure determination. *International Journal of High Performance Computing Applications*, 24(2):117–135, 2010.

[19] G. Desiderà, B. Anconelli, M. Bertero, P. Boccacci, and M. Carbillet. Application of iterative blind deconvolution to the reconstruction of LBT LINC-NIRVANA images. *Astron. Astrophys.*, 452:727–734, 2006.

[20] G. J. Hurford et al. The RHESSI imaging concept. *Solar Phys.*, 210(1/2):61, 2002.

[21] R. P. Lin et al. The Reuven Ramaty High-Energy Solar Spectroscopic Imager (RHESSI). *Solar Phys.*, 210(1/2):3, 2002.

[22] S. Krucker et al. Hard x-ray emission from the solar corona. *Astron. Astrophys. Rev.*, 16(3-4):155, 2008.

[23] D. A. Fish, A. M. Brinicombe, E. R. Pike, and J. G. Walker. Blind deconvolution by means of the Richardson-Lucy algorithm. *J. Opt. Soc. Am. A*, 12:58–65, 1995.

[24] J. Frank. *Three-Dimensional Electron Microscopy of Macromolecular Assemblies.* Oxford University Press, New York, 2006.

[25] G. H. Golub and V. Pereyra. Separable nonlinear least squares: the variable projection method and its applications. *Inverse Problems*, 19:R1–R26, 2003.

[26] G.H. Golub and V. Pereyra. The Differentiation of Pseudo-Inverses and Nonlinear Least Squares Problems Whose Variables Separate. *SIAM J. Numer. Anal.*, 10(2):413–432, 1973.

[27] E. Haber, U. M. Ascher, and D. Oldenburg. On optimization techniques for solving nonlinear inverse problems. *Inverse Problems*, 16(5):1263–1280, 2000.

[28] P. C. Hansen. Regularization tools: A Matlab package for the analysis and solution of discrete ill-posed problems. *Numerical Algorithms*, 6:1–35, 1994.

[29] P. C. Hansen. *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion.* SIAM, Philadelphia, 1998.

[30] P. C. Hansen. *Discrete inverse problems: insight and algorithms.* Fundamentals of algorithms. SIAM Society for industrial and applied mathematics, Philadelphia, 2010.

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

[32] Per Christian Hansen. Analysis of Discrete Ill-Posed Problems by Means of the L-Curve. *SIAM Review*, 34(4):561–580, 1992.

[33] R. J. Hathaway and J. C. Bezdek. Grouped coordinate minimization using Newton's method for inexact minimization in one vector coordinate. *Journal of Optimization Theory and Applications*, 71:503–516, 1991.

[34] M. Hohn, G. Tang, G. Goodyear, P. R. Baldwin, Z. Huang, P. A. Penczek, C. Yang, R. M. Glaeser, P. D. Adams, and S. J. Ludtke. SPARX, a new environment for Cryo-EM image processing. *J. Structural Biology*, 157(1):47–55, 2007.

[35] M. G. Kang and S. Chaudhuri. Super-resolution image reconstruction. *IEEE Signal Processing Magazine*, 20(3):19–20, 2003.

[36] L. Kaufman. A Variable Projection Method for Solving Separable Nonlinear Least Squares Problems. *BIT*, 15(1):49–57, 1975.

[37] L. Kaufman and V. Pereyra. A method for separable nonlinear least squares problems with separable equality constraints. *SINUM*, 15:12–20, 1978.

[38] C. T. Kelley. *Iterative Methods for Optimization.* SIAM, Philadelphia, 1999.

[39] R. Marabini, G. T. Herman, and J. M. Carazo. 3D reconstruction in electron microscopy using ART with smooth spherically symmetric volume elements (blobs). *Ultramicroscopy*, 72(1–2):53–65, 1998.

[40] A. M. Massone, A. G. Emslie, G. J. Hurford, E.P. Kontar M. Prato, and M. Piana. Hard X-ray imaging of solar flares using interpolated visibilities. *Astrophys. J.*, 703(2):2004, 2009.

[41] J. D. Monnier. Optical interferometry in astronomy. *Rep. Prog. Phys.*, 66: 789–857, 2003.

[42] V. A. Morozov. On the solution of functional equations by the method of regularization. *Soviet Math. Dokl.*, 7:414–417, 1996.

[43] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, New York, 1999.

[44] M.R. Osborne. Separable least squares, variable projection, and the Gauss-Newton algorithm. *Elec. Trans. Numer. Anal.*, 28:1–15, 2007.

[45] P. A. Penczek, M. Radermacher, and J. Frank. Three-dimensional reconstruction of single particles embedded in ice. *Ultramicroscopy*, 40(1):33–53, 1992.

[46] M. Piana, A. M. Massone, G. J. Hurford, M. Prato, A. G. Emslie, E. P. Kontar, and R. A. Schwartz. Electron flux spectral imaging of solar flares through regularized analysis of hard X-ray source visibilities. *Astrophys. J.*, 665(1):846, 2007.

[47] M. Prato. Regularization methods for the solution of inverse problems in solar X-ray and imaging spectroscopy. *Arch. Comput. Methods Eng.*, 16(2):109, 2009.

[48] L. I. Rudin and S. Osher. Total Variation based image restoration with free local constraints. *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, 1:31–35, 1994.

[49] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear Total Variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[50] A. Ruhe and P. Wedin. Algorithms for Separable Nonlinear Least Squares Problems. *SIAM Review*, 22(3):318–337, 1980.

[51] S. D. Saban, M. Silvestry, G. R. Nemerow, and P. L. Stewart. Visualization of α-helices in a 6-Ångstrom resolution cryoelectron microscopy structure of adenovirus allows refinement of capsid protein assignments. *J. Virol*, 80(24):49–59, 2006.

[52] D. M. Sima and S. Van Huffel. Separable nonlinear least squares fitting with linear bound constraints and its application in magnetic resonance spectroscopy data quantification. *J. Comput. Appl. Math.*, 203:264–278, 2007.

[53] D.L. Snyder, A. M. Hammoud, and R. L. White. Image recovery from data acquired with a charge-coupled-device camera. *J. Opt. Soc. Am. A*, 10:1014–1023, 1993.

[54] D.L. Snyder, C.W. Helstrom, A.D. Lanterman, M. Faisal, and R. L. White. Compensation for readout noise in CCD images. *J. Opt. Soc. Am. A*, 12:272–283, 1995.

[55] C. R. Vogel. *Computational Methods for Inverse Problems.* SIAM, Philadelphia, 2002.

[56] Y. Xu, A. G. Emslie, and G. J. Hurford. RHESSI hard X-ray imaging spectroscopy of extended sources and the physical properties of electron acceleration regions in solar flares. *Astrophys. J.*, 673: 576–585, 2008.

[57] Y. You and M. Kaveh. A regularization approach to joint blur identification and image restoration. *IEEE Trans. Image Process.*, 5(3):416–428, 1996.

[58] Y. You and M. Kaveh. Blind image restoration by anisotropic regularization. *IEEE Trans. Image Process.*, 8(3):396–407, 1999.