Università degli Studi di Modena e Reggio Emilia

School of Graduate Studies
Multiscale Modelling, Computational Simulations and Characterization in Material and Life Sciences
- XXVI Cycle -

METHODS AND SOFTWARE FOR IMAGE RESTORATION ON DIFFERENT PARALLEL ENVIRONMENTS

by

Roberto Cavicchioli

Doctoral Thesis

2013

Acknowledgement

Foremost, I would like to express my sincere gratitude to my advisor Prof. Luca Zanni for the continuous support of my Ph.D study and research, for his patience, enthusiasm, and great humanity.

I would like also to express my sincere gratitude to Dr. Marco Prato for his encouragement, energy and patience during my Ph.D period.

My sincere thanks also go to Prof. Laure Blanc-Féraud and Dr. Caroline Chaux for offering me the internship opportunities in the I3S group at University of Nice-Sophia Antipolis and for leading me working on an exciting research project.

I also would like to thank all the brilliant people with whom I had the privilege to collaborate: Prof. Mario Bertero, Prof. Patrizia Boccacci, Dr. Riccardo Zanella, Prof. Gaetano Zanghirati, Prof. Giuseppe Vicidomini and Dr. Andrea Prearo.

I thank my fellow labmates Anastasia Cornelio and Federica Porta, for the stimulating discussions, for mutual support when we were working together before deadlines, and for all the fun we have had in the last three years.

Last but not the least, I would like to thank my family: my parents Claudio Cavicchioli and Patrizia Rinaldi and my brother Paolo Cavicchioli, for supporting me spiritually throughout my life, and the love of my life, Martina Aldrovandi.

Contents

List of Algorithms

Introduction

Since many years, digital images have been used in various areas of science, such as astronomy, biology and medicine, but in all the cases these are unavoidably corrupted by noise and blur.

In particular the noise is mainly due to the conversion from analogical to digital signal, while the blur usually results from the nature of the observations, such as atmosphere for astronomy or lenses aberration in the optical system.

Images obtained by confocal microscopy, for example, are contaminated by blur and noise, caused mainly by aberration in the optical system, light diffraction and conversion of data from analogical signals to digital ones.

The algorithms based on a statistical approach are the most useful for the restoration of this kind of images. These methods are called Maximum Likelihood (ML) methods and are well described in [13]. All the algorithms belonging to this class assume that the input image suffers from noise satisfying to a Gaussian or Poisson distribution and provide a reconstruction by solving an optimization problem corresponding to the application of an ML criterion. Although statistical deconvolution methods usually give acceptable results in terms of reconstruction accuracy, they are often computationally intensive. One possible solution to overcome this problem is the parallelization of the computation for exploiting up-to-date multiprocessor architectures. The most common parallelization strategies currently exploited may be categorized into two main categories.

The first category includes algorithms that are implemented as multi-threaded applications and are developed for symmetric multiprocessing architectures (SMP). This model assumes that all processing units (processors) are identical and use single shared main memory. Such parallel application can use nowadays PC machines which multi-core processors, workstations with several processors and shared global memory, or a different kind of or high-performance multiprocessors

devices such as Graphics Processing Units (GPU).

The algorithms belonging to the second category are designed for distributed memory architectures. In this case, each processor has its own private memory. Computational tasks can only operate on local data and must communicate with other tasks to access their data. Software developed with this approach can use a computer cluster (a group of network-linked PC machines or workstations) and can also efficiently work on SMP architectures.They can use SMP to speed the computation by splitting the calculation among threads.

However, deconvolution of large images may be still a problem. Standard computers usually contain only a few GBs of memory and one processor with two or four cores. Multiprocessor workstation are expensive and contain no more than four CPUs. Some GPU architectures could be a good solution to achieve the best speed, because of their massive parallel structure.

In this thesis we propose two parallel versions of the Scaled Gradient Projection (SGP) method based on an adaptive alternation of the Barzilai–Borwein rules for updating the steplengths parameter and a special strategy for scaling the gradient direction. Furthermore, an extension to consider also a correction for the reduction of the boundary effects typically introduced by the reconstruction algorithms is developed. The first parallel version uses the Message Passing Interface (MPI) environment and works efficiently on clusters of computers; the other uses the NVidia CUDA framework for exploiting GPU devices. The implementation was designed principally for 2-dimensional cases, but it has been extended to work for N-dimensional images.

This thesis is organized as follows.

In Chapter 1 we introduce the imaging problems arising from astronomical and microscopy fields.

In Chapter 2 we present the SGP method, then we propose the adaptive steplength rule adopted in SGP and the strategy for scaling the gradient direction. Finally an extension to give models to solve reconstruction artifacts, like boundary effects, and for restoration from multiple images are described.

Chapter 3 deals with the implementation of the algorithm on distributed memory architecture and GPU-computing; both the MPI and the NVidia CUDA framework are adopted.

Chapter 4 describes another computationally intensive problem in imaging:

the regularization hyperparameter estimation in inverse problem with wavelet regularization. We introduce an accelerated gradient method based on the same SGP steplength rule for solving the optimization problem arising in an ML estimation strategy.

Chapter 5 reports the numerical experiment showing the improvements achievable with the SGP parallel implementation and the effectiveness of the proposed gradient method for the ML estimation of the regularization hyperparameter.

Chapter 1

Imaging Problems

Image reconstruction is one of the main topic of imaging problems; in particular we can assume that images are mainly corrupted by blurring and noise. Image deblurring is a linear inverse problem since it consists in the inversion of a linear and continuous operator $A : X \to X$ where X is, for instance, a Hilbert space of square-integrable functions. In many instances, A is a convolution operator, i.e. $Ax = K * x$ for any $x \in X$, K being an integrable function, the so-called Point Spread Function (PSF) of the imaging system. By representing a two dimensional image $\mathcal{X} \in \mathbb{R}^{n \times n}$ as a vector $x = (x_1, \ldots, x_N)^T \in \mathbb{R}^N, N = n^2$, in which the entries of $\mathcal{X}$ are stacked column-wise, we can write

$$Ax = y \tag{1.1}$$

where x is the image to be reconstructed and $y \in \mathbb{R}^N$ is the sum of two terms: $y = g + \eta$; here $g \in \mathbb{R}^N$ is the blurred image that has been recorded in absence of noise and $\eta \in \mathbb{R}^N$ is the noise affecting the image acquisition [13]. The image restoration problem is then to obtain an approximation of x knowing A and y (see Figure 1.1).

Since the system (1.1) is given by discretization of an ill-posed problem, the matrix A could be very ill-conditioned and a conventional approach to the solution of the system is in general not successful. Consequently alternative strategies must be exploited, usually in the form of iterative schemes motivated by different approaches, like solving linear equations or variational problems [119].

One of the principal ways to face the problem is considering a constrained

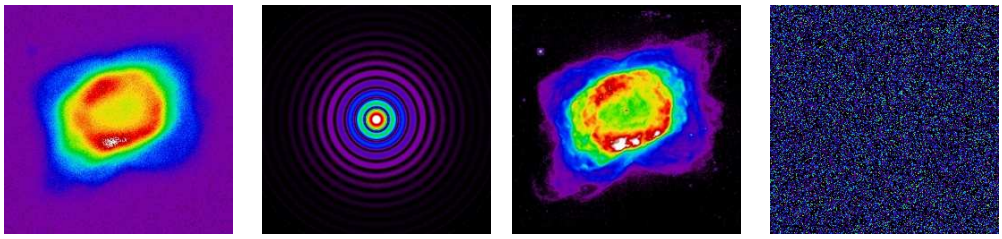


Figure 1.1: Example of an imaging restoration problem.

minimization problem in this form:

$$\begin{aligned} &\min \quad f(\boldsymbol{x}) \\ &\text{sub. to} \quad \boldsymbol{x} \in \Omega, \end{aligned} \tag{1.2}$$

where $\Omega \subset \mathbb{R}^N$ is a closed convex set and $f : \Omega \to \mathbb{R}$ is a continuously differentiable function. We are interested in the case where the feasible region Ω is described by simple constraints. This situation arises for example in many imaging problems where the constraints are related to the non-negativity of the solution, (e.g. positive values of pixels)

$$\Omega = \left\{ \boldsymbol{x} \in \mathbb{R}^N \mid x_i \geq 0, \quad \forall\, i = 1, \ldots, N \right\}, \tag{1.3}$$

or, they can be used to force also the so-called flux conservation property,

$$\Omega = \left\{ \boldsymbol{x} \in \mathbb{R}^N \mid x_i \geq 0, \quad \forall\, i = 1, \ldots, N, \quad \sum_{i=1}^{N} x_i = c \right\}, \tag{1.4}$$

where c is a prefixed positive constant.

If possible f is formed by a couple of terms:

$$f(\boldsymbol{x}) = f_0(\boldsymbol{x}) + \lambda f_R(\boldsymbol{x}) \tag{1.5}$$

where the first term f_0 is called *data-fidelity* term and measures the difference between reconstructed and measured data, while f_R is the *penalty* term or *regularization* term and encodes additional information about the solution. Gradient

projection type methods are appealing for these problems for a couple of reasons. Firstly, the particular structure of the constraints makes the projection of a vector on the feasible region an operation with a considerably low computational cost; for the case (1.3) is obvious, and for case (1.4) algorithms with linear computational cost have been developed [38]. Secondly, the recent advances on the steplength selection in gradient methods [37, 36, 129, 53] permit to greatly improve the convergence speed of these schemes without the introduction of an expensive computational cost. Thus, new gradient projection methods can nowadays be designed that, thanks to the low computational cost per iteration and the good convergence rate, may represent a valid alternative to other gradient-based iterative approaches widely used in image restoration [39, 84, 90, 101, 106].

1.1 Digital image restoration and optimization

Since this work is focused on digital image reconstruction, it is useful to explicitly mention the link to the kind of first-order optimization algorithms we consider here.

Using standard naming and notation, we call "object" the unknown entity $\boldsymbol{x}$ we want to reconstruct, whose observations are the known data $\boldsymbol{y}$, and, after discretization, we always suppose to have reordered both the object and the data arrays so that $\boldsymbol{x}$ and $\boldsymbol{y}$ be column vectors. As it is well known, the image reconstruction is an ill-posed inverse problem, where usually the main difficulty is the lack of continuous dependency of the solution $\boldsymbol{x}^*$ on the data. After discretization, the ill-posedness implies that a discrete ill-conditioned problem has to be solved. The most widely used approaches to the solution of the reconstruction problem are based on the well known Tikhonov's regularization theory. However, despite the huge amount of available literature and the deep results known in this theory, the *statistical approaches* are recently receiving increasing attention. They are related to the idea of modelling the data acquisition as a random process, so that the measured data $\boldsymbol{y}$ are realization of a random variable $Y \in \mathbb{R}^N$. This viewpoint has a number of relevant benefits: the noise affecting the recorded data as well as a possible *background* radiation are included in a natural way, different kind of noise can be easily modelled and, very important, it naturally allows the inclusion in the problem of *a priori* information on its solution. As it is well known, the

latter is the key point to remove ill-posedness and to obtain meaningful results, whatever is the chosen approach to the problem.

Hence, one can look at *statistical methods* to obtain a reasonable estimate of the unknown object $\boldsymbol{x}$ to be reconstructed. In this context, the two main classes are the *maximum likelihood* (ML) and the *maximum a posteriori* (MAP) estimation techniques. The former is connected to the *parameter estimation* idea, while the latter is based on the Bayes formula.

In the ML case, it is assumed that the data probability density function $p_Y(\boldsymbol{y};\boldsymbol{x})$ is known: this function is then used as a measure of the closeness to $\boldsymbol{y}$ of a given $\boldsymbol{x}$, that is, of their *likelihood*. Hence, one looks for a point $\boldsymbol{x}^*$ *maximizing the likelihood* to $\boldsymbol{y}$, that is a solution of

$$\max_{\boldsymbol{x}\in\mathbb{R}^N} L_{Y,\boldsymbol{y}}(\boldsymbol{x}) \qquad \text{with} \qquad L_{Y,\boldsymbol{y}}(\boldsymbol{x}) = p_Y(\boldsymbol{y};\boldsymbol{x})\,. \tag{1.6}$$

In the digital image reconstruction context, reasonable assumptions allows to consider the *loglikelihood* function $f_0(\boldsymbol{x};\boldsymbol{y}) = -\gamma_1 \ln\big(L_{Y,\boldsymbol{y}}(\boldsymbol{x})\big) + \gamma_2$ (with $\gamma_1, \gamma_2 \in \mathbb{R}$, $\gamma_1 \neq 0$), that makes (1.6) equivalent to

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} f_0(\boldsymbol{x};\boldsymbol{y})\,. \tag{1.7}$$

The actual form of f_0 depends on the density $p_Y(\boldsymbol{y};\boldsymbol{x})$: it usually models the kind of noise affecting the data and results in a (possibly highly) nonlinear f_0. Well known examples are the least squares function for Gaussian noise, the Kullback-Leibler (KL) divergence for Poisson noise and a sum of nonlinear functions for the combination of the two. Usually, one looks for non-negatively constrained solutions of (1.7), since pixel intensity is within a non-negative range.

The MAP estimate is based on the Bayesian approach, which considers also the object $\boldsymbol{x}$ as a realization of a random variable $X \in \mathbb{R}^N$. This means, in turn, that $p_Y(\boldsymbol{y};\boldsymbol{x})$ becomes the *conditional probability density function* of the data $\boldsymbol{y}$ given the object $\boldsymbol{x}$. It is usually written as $p_Y(\boldsymbol{y}|\boldsymbol{x})$. In this context, inverting the acquisition process means to determine the *a posteriori* probability density function $p_X(\boldsymbol{x}|\boldsymbol{y})$, that is the density of X given the observed data $\boldsymbol{y}$. One can then naturally include known additional information on the solution via the probability density $p_X(\boldsymbol{x})$, also said *the prior*. From the Bayes formula $p_X(\boldsymbol{x}|\boldsymbol{y})p_Y(\boldsymbol{y}) = p_Y(\boldsymbol{y}|\boldsymbol{x})p_X(\boldsymbol{x})$, given the *marginal probability density* $p_Y(\boldsymbol{y})$ as a

function of the recorded data $\boldsymbol{y}$, one can finally get an estimate of the unknown object by *maximizing* the a posteriori probability for the given data, that is by solving

$$\max_{\boldsymbol{x}\in\mathbb{R}^N} P_{X,\boldsymbol{y}}(\boldsymbol{x}) \qquad \text{with} \qquad P_{X,\boldsymbol{y}}(\boldsymbol{x}) = L_{Y,\boldsymbol{y}}(\boldsymbol{x}) p_X(\boldsymbol{x})/p_Y(\boldsymbol{y})\,. \tag{1.8}$$

It is clear that the actual form of $P_{X,\boldsymbol{y}}(\boldsymbol{x})$ depends on the form of $p_X(\boldsymbol{x})$. If one can assume that such a function can be written as $p_X(\boldsymbol{x}) = \gamma \exp(-\mu h(x))$, $\gamma, \mu > 0$, then by applying to the objective function the same logarithmic transformation as for ML one gets the equivalent problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} f(\boldsymbol{x};\boldsymbol{y}) \qquad \text{with} \qquad f(\boldsymbol{x};\boldsymbol{y}) = f_0(\boldsymbol{x};\boldsymbol{y}) + \lambda f_R(\boldsymbol{x}) \tag{1.9}$$

where $f_R(\boldsymbol{x}) = \gamma_1 h(\boldsymbol{x})$ and in the objective function we have neglected the constant term $r = \gamma_1\Big(\ln\big(p_Y(\boldsymbol{y})\big) - \ln(\gamma)\Big)$. The function f_R is called *regularizer* and the parameter λ is the *regularization parameter*.

Thus, the image restoration problem can be formulated as an optimization problem of the form (1.2)–(1.3) in which the objective function is as (1.7) or (1.9). It is important to remark that in the case of the objective function (1.7), which does not include any prior information, regularized solutions of the ill-conditioned reconstruction problem are usually obtained by early stopping suited iterative minimization methods.

Since it is often the case in imaging applications that $f(\boldsymbol{x};\boldsymbol{y})$ is convex, the problem (1.2)–(1.3) becomes convex, so all its solutions are global minimizers.

1.2 Data affected by Poisson Noise

In case of deconvolution methods for image deblurring, the Richardson-Lucy (RL) [101, 90] method is the most popular in Astronomy because it preserves the number of counts and the non-negativity of the original object. It is an iterative method (also known as Expectation Maximization (EM) in other scientific environments) in which the iteration, as modified by Snyder [108], is

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} A^T \frac{\boldsymbol{y}}{A\boldsymbol{x}^{(k)} + \boldsymbol{b}} \tag{1.10}$$

if the PSF is normalized to unit volume, where A^T is the transposed matrix, $\boldsymbol{b}$ is a known array representing background emission, and the quotient of two arrays is intended as Hadarmard operations, that is componentwise operations. Here it is assumed that A has only non-negative elements and that all its rows and all its columns have at least one nonzero element.

Regularization is usually obtained by an early stopping of RL iterations instead of adding a *penalty* term; in the case of point-wise objects such as binaries or open star clusters, iterations can however be pushed to convergence.

The main disadvantage of the RL algorithm is that it is not very efficient: it may require hundreds or thousands of iterations for images with a large number of counts (low Poisson noise). In the case of large scale images or multiple images of the same target, the computational cost can become prohibitive. For this reason several acceleration schemes have been proposed. Many of them are based on the remark that RL can be seen as a scaled gradient method for the minimization of the Kullback-Leibler (KL) divergence, which represents the *data-fidelity* function in the case of Poisson noise:

$$f_0(\boldsymbol{x};\boldsymbol{y}) = \sum_{\mathbf{m}\in S} \left\{ \boldsymbol{y}(\mathbf{m}) \ln \frac{\boldsymbol{y}(\mathbf{m})}{(A\boldsymbol{x})(\mathbf{m}) + \boldsymbol{b}(\mathbf{m})} + (A\boldsymbol{x})(\mathbf{m}) + \boldsymbol{b}(\mathbf{m}) - \boldsymbol{y}(\mathbf{m}) \right\} \quad , \tag{1.11}$$

where S is the set of values of the multi-index $\mathbf{m}$ labelling the image pixels.

The method has been deeply investigated and modified to improve speed of convergence. First there is the "multiplicative relaxation" proposed by Llacer & Núñez [89] which consists in replacing the iteration (1.10) by the following one

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} \left(A^T \frac{\boldsymbol{y}}{A\boldsymbol{x}^{(k)} + \boldsymbol{b}} \right)^{\alpha} \tag{1.12}$$

with $\alpha > 1$. Convergence is proved in Iusem [78] for $\alpha < 2$. As shown in Lantéri et al [85] this approach can provide a reduction in the number of iterations by a factor α, with essentially the same cost per iteration. For low numbers of counts numerical convergence has been found also for $\alpha > 2$ (Anconelli et al. [3]). Secondly a "linear relaxation" is investigated in Adorf et al. [1]. It can be written in

the following form

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \lambda_k \boldsymbol{x}^{(k)} \left(\mathbf{1} - A^T \frac{\boldsymbol{y}}{A\boldsymbol{x}^{(k)} + \boldsymbol{b}} \right) \quad , \tag{1.13}$$

where $\lambda_k > 1$ (for $\lambda_k = 1$ the RL algorithm is re-obtained) and $\mathbf{1}$ is the array with all entries equal to 1. It can be noticed that the quantity in brackets is the gradient of $f_0(\boldsymbol{x}; \boldsymbol{y})$, therefore RL is a scaled gradient method with a scaling given by $\boldsymbol{x}^{(k)}$ at iteration k, and that the relaxation method is essentially a line-search along this descent direction; it can be performed by minimizing the objective function $f_0(\boldsymbol{x}; \boldsymbol{y})$ (Adorf et al. [1]) or by using Armijo rule (Lantéri et al. [85]). A moderate speedup is observed by these authors. Convergence of the algorithms is derived from general results in optimization theory (Bertsekas [18]). Finally, higher speedup, of the order of 10 or more, is observed using an acceleration method proposed by Biggs & Andrews [19], which exploits a suitable extrapolation along the trajectory of the iterates, and is implemented in the *deconvlucy* function of the *Image Processing* MATLAB toolbox. The problem with this method is that no convergence proof is available and, in our experience, a deviation from the trajectory of RL iterations is sometimes observed, providing unreliable results.

In a recent paper [21] a general optimization method, called Scaled Gradient Projection (SGP) method, has been proposed for the constrained minimization of continuously differentiable convex functions. It is applicable to the non-negative minimization of the KL divergence. If the scaling suggested by RL is used in this method, then it provides a considerable speedup of the algorithm. The performance of the new method is compared with those of RL and of the Biggs & Andrews method, as implemented in MATLAB, obtaining a speedup comparable with that of the latter method, and sometimes better, without its drawbacks. Further applications of SGP in image restoration problems can be found e.g. in Benvenuto et al. [10], Bonettini & Prato [22], Zanella et al. [124].

The main feature of the gradient method proposed in [23] consists in the combination of non-expensive diagonally scaled gradient direction with steplength-selection rules specially designed for these directions. Also, global convergence is ensured by the usage of a nonmonotone line-search strategy along the feasible direction. Further explanation of the method is given in Chapter 2

When we want to solve an image denoising problem, the modelling is the

same. We solve (1.2)–(1.3) with the objective function (1.9), where f_0 has the form (1.11) with $A = I_N$ and the regularizer $f_R(\boldsymbol{x})$ can be chosen as the approximated *total variation* (TV) functional described in [124]. In this functional, the non-differentiable integrand function $|\nabla x|$ of the standard TV regularizer is approximated by $\sqrt{t+\delta^2}$, where $t = \|\mathcal{D}^2(\boldsymbol{x})\|^2$ is the squared norm of a discrete derivative operator $\mathcal{D}$ and $\delta > 0$ is a (small) smoothing parameter. Here the SGP method is used as a standard minimization algorithm and it is stopped when the relative difference on function values crosses a given tolerance.

1.3 Image restoration in astronomy

When we want to obtain an image from an astronomical telescope, but when astronomical objects are observed from the Earth, the atmosphere distorts the light and the resulting image suffers from blurring. Also, due to the conversion from analogical to digital signals, noise is present.

To model the distortion we can proceed with an empirical method, that consists in extracting the PSF directly from the observation of a single star, or from a theoretical model. In the second case the model has been constructed following Goodman [59] and implemented in the software package AIRY [25, 26].

In this thesis we focused on a particular astronomical imaging problem, arising from the reconstruction of images taken with the Large Binocular Telescope. The Large Binocular Telescope (LBT) has been designed for obtaining optical and infrared images with high sensitivity and resolution. It consists of two 8.4m mirrors on a common mount, with a spacing of 14.4m between the centers. Adaptive optics counteracts the blurring effect of atmospheric turbulence whereas interferometry improves the resolution of a single aperture (see Figure 1.2).

For a given orientation of this telescope, the diffraction-limited resolution along the center-to-center baseline is equivalent to that of a 22.8m mirror, while the resolution in the perpendicular direction is that of a standard 8m mirror. However, the rotation of the baseline induced by the Earth motion during the night and along the year can make accessible many different parallactic angles for each target so that, in principle, a coverage equivalent to that of a 22.8m mirror can be obtained.

LBT has been designed to offer astronomers as much flexibility as possible. The two large mirrors can be used separately with different instruments above

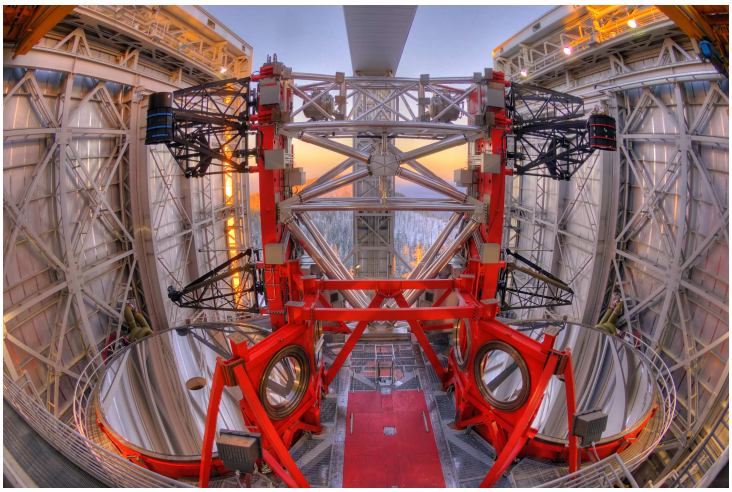


Figure 1.2: A photo of the LBT located near Safford, Arizona.

each. These are swung into position on gantries and allow different aspects of the same object to be measured simultaneously. However, the light from each mirror can also be combined to make one high-resolution image. In Figure 1.3 are shown these parts:

1. As the light enters the telescope and hits the huge mirrors, the largest ever made from a single piece of glass, it is deflected towards the secondary mirrors above.

2. Because the Earth's atmosphere distorts the light from stars, adaptive optics are used to correct the image. Moving magnets on the back of the secondary mirrors can change their shapes 1000 times second.

3. The corrected light is deflected towards the centre of the telescope where one of various instruments combines the beams.

4. The dynamic balancing system compensates for movements of the telescope and helps it remain fixed on one spot in space.

5. The light can be used in different ways. The LBT interferometer can be used to reverse its phase, effectively cancelling the light from a bright star and allowing astronomers to look for faint, orbiting planets.

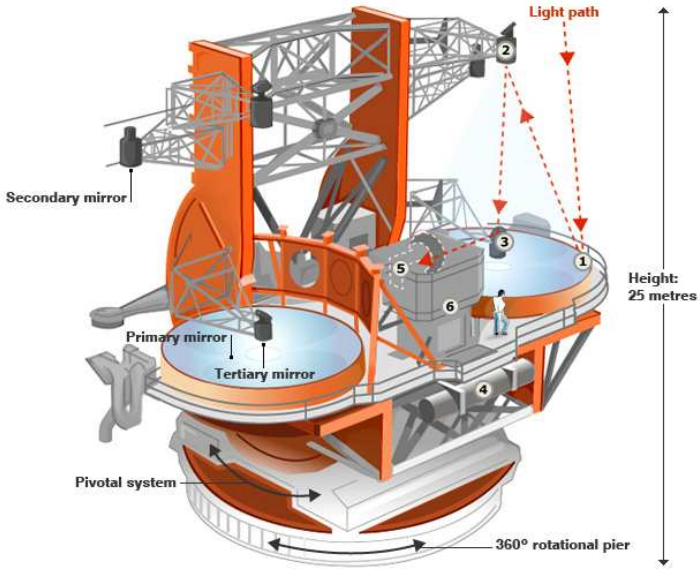


Figure 1.3: A scheme of the LBT working steps.

6. The interferometric camera is used to combine the light in phase, allowing high-resolution images to be taken. In this mode, the telescope has the equivalent sharpness of a 22.8m instrument.

In particular LINC-NIRVANA (LN) is an adaptive optics near-infrared Fizeau imaging interferometer that combine the beams from the two primary mirrors of the LBT (Herbst et al. [71]). LN has been developed by a Consortium of German and Italian institutes, and it has been installed at the LBT instrument platform at the lower part of the area between the two mirrors.

This interferometer permits to achieve the higher resolution images, but its PSF corresponds to the Airy pattern of a single 8.4m telescope superimposed by interference fringes created by the operation of two 8.4m telescopes in Fizeau interferometric mode. Due to the maximum optical extent of 22.8m (the line connecting the extreme points of both telescope primary mirrors which determines the highest accessible spatial frequency) approximately three fringe maxima fit into the central maximum of the 8.4m Airy distribution.

LN will be equipped with a detector consisting of 2048 $\times$ 2048 pixels with a pixel size of about 5mas, corresponding to a F0V of 10" $\times$ 10" for each orientation

of the baseline. Since in K-band the resolution of a 22.8m mirror is about 20mas, the detector provides an oversampling by a factor 4.

Because of the sky rotation, the detector must constantly be de-rotated during an exposure in order to avoid image blurring. However, this causes a changing orientation of the telescope baseline in the image (which would otherwise remain stable). While the distribution of objects on the detector is maintained, the PSF rotates, and this fact leads to a smearing out of the interference fringes and thus to a loss of fringe contrast and spatial resolution. To overcome this fringe contrast we can reconstruct images taken at different times of the same object, but with different PSFs orientations.

1.3.1 Single image deconvolution

The problem of single image deconvolution with photon counting noise is the minimization of the KL divergence defined in Eq. (1.11) and the solution can be obtained by applying the iterative RL algorithm of Eq. (1.10). For applying SGP we only need the expression of the gradient of the KL divergence which is given by (the normalization of the PSF to unit volume is used)

$$\nabla f_0(\boldsymbol{x};\boldsymbol{y}) = \mathbf{1} - A^T \frac{\boldsymbol{y}}{A\boldsymbol{x}+\boldsymbol{b}} \quad . \tag{1.14}$$

The SGP behaviour with respect to RL is investigated in Bonettini et al.[23].

1.3.2 Multiple images deconvolution

The problem of multiple image deconvolution is basic for the Fizeau interferometer of LBT or for the "co-adding" method of images with different PSFs proposed by Lucy & Hook [91].

Let p be the number of detected images $\boldsymbol{y}_j$, (j=1,..,p), with corresponding PSFs $\boldsymbol{K}_j$, all normalized to unit volume, and let us set $A_j\boldsymbol{x} = \boldsymbol{K}_j * \boldsymbol{x}$. It is quite natural to assume that the p images are statistically independent so that the likelihood of the problem is the product of the likelihoods of the different images. If we assume again Poisson statistics, by taking the negative logarithm of the likelihood, then the maximization of the likelihood is equivalent to the minimization of a data-fidelity

function which is the sum of KL divergences, one for each image, i.e.

$$f_0(\boldsymbol{x};\boldsymbol{y}) = \sum_{j=1}^{p} \sum_{\mathbf{m}\in S} \{\boldsymbol{y}_j(\mathbf{m}) \ln \frac{\boldsymbol{y}_j(\mathbf{m})}{(A_j\boldsymbol{x})(\mathbf{m}) + \boldsymbol{b}_j(\mathbf{m})} + \\ + (A_j\boldsymbol{x})(\mathbf{m}) + \boldsymbol{b}_j(\mathbf{m}) - \boldsymbol{y}_j(\mathbf{m})\} \quad . \tag{1.15}$$

If we apply the standard expectation maximization method (Shepp & Vardi [106]) to this problem, we obtain the iterative algorithm

$$\boldsymbol{x}^{(k+1)} = \frac{1}{p}\boldsymbol{x}^{(k)} \sum_{j=1}^{p} A_j^T \frac{\boldsymbol{y}_j}{A_j\boldsymbol{x}^{(k)} + \boldsymbol{b}_j} \quad , \tag{1.16}$$

which we call the *multiple image* RL method (multiple RL, for short). Since the gradient of (1.15) is given by

$$\nabla f_0(\boldsymbol{x};\boldsymbol{y}) = \sum_{j=1}^{p} \left\{ \mathbf{1} - A_j^T \frac{\boldsymbol{y}_j}{A_j\boldsymbol{x} + \boldsymbol{b}_j} \right\} \quad , \tag{1.17}$$

we find that the algorithm (1.16) is a scaled gradient method, with a scaling given, at iteration k, by $\boldsymbol{x}^{(k)}/p$. Therefore the application of SGP to this problem is straightforward.

However, for the reconstruction of LINC-NIRVANA images, we must consider that an acceleration of the algorithm (1.16) is proposed in Bertero & Boccacci [11] by exploiting an analogy between the images of the interferometer and the projections in tomography. In this approach called OSEM (ordered subset expectation maximization, Hudson & Larkin [74]) the sum over the p images in Eq. (1.16) is replaced by a cycle over the same images. In order to avoid oscillations of the reconstructions at the interior of the cycle, a preliminary step is the normalization of the different images to the same flux, if different integration times are used in the acquisition process. OSEM is summarized in Algorithm 1.1.

As follows from practice and theoretical remarks, this approach reduces the number of iterations by a factor p. However the computational cost of one multiple RL iteration is lower than that of one OSEM iteration: we need $3p + 1$ FFTs in the first case and $4p$ FFTs in the second one. In conclusion, the speedup provided by OSEM is roughly given by $(3p+1)/4$. If $p = 3$ (the number of images provided by the interferometer will be presumably small) the speedup is 2.5 and increases

ALGORITHM 1.1 Ordered Subset Expectation Maximization (OSEM) method

Choose the starting point $\boldsymbol{x}^{(0)} > 0$.

FOR $k = 0, 1, 2, ...$ DO THE FOLLOWING STEPS:

1. Set $\boldsymbol{h}^{(0)} = \boldsymbol{x}^{(k)}$;
2. FOR $j = 1, ..., p$ COMPUTE

$$\boldsymbol{h}^{(j)} = \boldsymbol{h}^{(j-1)} \left(A_j^T \frac{\boldsymbol{y}_j}{A_j \boldsymbol{h}^{(j-1)} + \boldsymbol{b}_j} \right) \quad ; \tag{1.18}$$

3. Set $\boldsymbol{x}^{(k+1)} = \boldsymbol{h}^{(p)}$.

END

to 4.7 if $p = 6$. These results must be taken into account for appreciating the speedup of SGP with respect to multiple RL. We can add that the convergence of SGP is proved while that of OSEM is not, even if it has been always verified in our numerical experiments.

1.4 Image restoration in microscopy

Image deconvolution is a computational technique that mitigates the distortions created by any optical system, therefore is possible to apply it also in the microscopy field. Agard first applied image deconvolution to fluorescence microscopy in the early 1980s [2]. In this paper Agard proposed different algorithms for deconvolving images acquired as three-dimensional (3D) stacks using wide-field microscopy (WFM). To describe the process naively, the focal plane of the objective lens moves along the thickness of the specimen and, for each position, the microscope generates a bi-dimensional (2D) image. Due to the diffraction phenomena, each 2D image, also called optical section, includes considerable out-of-focus light originating from regions of the specimen above and below the focal plane. Image deconvolution uses information describing how the microscope produces the image (forward model) as the basis of a mathematical transformation that reassigns the out-of-focus light to the points of origin.

Later, many new optical methods have been proposed to remove out-of-focus light and to generate directly true optical sections. Examples of these methods

are confocal laser scanning microscopy (CLSM) [45, 98], two-photon excitation microscopy (TPEM) [45, 46] and selective plane illumination microscopy (SPIM) [76, 75]. All of these remove out-of-focus light by rejecting such light before it reaches the detector or by precluding its generation. Further hybrid techniques, which remove out-of-focus light by combining optical and computational methods are 4Pi microscopy [70, 117] and structured illumination microscopy (SIM) [96, 63].

Since CLSM, TPEM and SPIM have considerably smaller contribution of out-of-focus light they are sometimes considered as pure alternatives to the deconvolution and WFM combination use. However, it has been shown that also these techniques can strongly benefit from image deconvolution [14, 93, 111, 94]. Although out-of-focus background is reduced, the images produced by such systems are still blurred versions of the specimen's structures in the focal plane and are contaminated by noise, thereby deconvolution can improve their contrast and signal-to-noise ratio. Similarly, also single 2D image can benefit of deconvolution, especially when obtained from thin specimen, where out-of-focus background vanishes.

More recently new super-resolution fluorescence microscopy approaches (usually referred to as nanoscopy) have enlarged the portfolio of tools for investigating biological samples [73]. The nanoscopy techniques have effectively broke the diffraction barrier and moved the spatial resolution of fluorescence microscopy down to the nanoscale [69]. Nevertheless, also in these cases image deconvolution can help to improve the quality of resulting images. This has been demonstrated both for stimulated emission depletion (STED) microscopy [48], which at moment can be considered as the method of choice between the targeted nanoscopy techniques, and, more recently, also for stochastic nanoscopy techniques [95]. It can be stated that all microscopy techniques including directly or indirectly a convolution in their image formation processes can benefit from image deconvolution. It is also important to remember that any quantitative analysis on fluorescence images, e.g. colocalization analysis or volume/area estimations, are significantly improved if performed on deconvolved images [47, 113].

In this scenario, one expects that any 2D or 3D image obtained from almost any fluorescence microscope is deconvolved before being analyzed. This unfortunately is not true, because the main disadvantage that precludes this massive spreading of deconvolution is the high computational demand which leads to long waiting time before producing the result. As a consequence in many applications

image deconvolution is not used to avoid strong delay in the data analysis pipeline. The situation becomes almost prohibitive in the case of large-scale images. For the above mentioned reasons, several methods to increase the speed of the deconvolution process have been proposed.

Two main directions have been followed. The first one relays on the implementation of the algorithms, *i.e.*, parallelization of the calculus and/or implementation on graphics processing units (GPUs) [87, 103, 104, 24]. A second approach, which found a strong attraction earlier, relays on the development of schemes to accelerate the deconvolution algorithms [19, 89]. Even if linear deconvolution like the usage of Wiener filtering, is extremely fast, its application to noisy images provides in general poor results; on the other side nonlinear deconvolution methods, and in particular iterative methods (with or without regularization), lead to excellent results but their convergence is very slow, requiring hundreds or thousands of iterations. The major representative algorithms for nonlinear deconvolution in fluorescence microscopy are based on the maximum-likelihood (ML) approach and, for the regularized version, on the maximum a posteriori (MAP) approach [15]. These algorithms can take advantage from prior information about the image formation process and the specimen, effectively reducing the ill-posedness of the problem. Most of this algorithms are iterative first-order methods, hence their implementation is easy (basically computation of a matrix-vector multiplication at each iteration), but, as already mentioned above, their convergence is very slow.

In the case of a differentiable penalty function $f_R(x)$ several iterative methods have been proposed for the minimization of the function $f(x)$. Amongst all, two methods are interesting: the one-step late (OSL) method proposed in [60] and the split-gradient method (SGM) proposed in [84]. The first is used for instance in [44] for total variation (TV) regularization and the second in [112] for Markov random field (MRF) regularization.

It is easy to show that both OSL and SGM are scaled gradient methods; however only in the case of SGM the scaling is non-negative for any regularization function $f_R(x)$ and any value of the regularization parameter. Therefore our reference algorithms are RL for the maximum-likelihood approach and SGM for the Bayes approach.

In the first case SGP is able to provide a very efficient solution of the ML image deconvolution, hence an acceleration of the RL method; in the second case

an efficient solution of the MAP image deconvolution with $f_R(x) = ||x||_2^2$, hence an acceleration of the algorithm proposed in [34].

The SGP algorithm mentioned in (1.2) can be exploited in this case and also used to derive an acceleration for another important widely used regularized deconvolution algorithm based on a quadratic regularization term [34].

It has been shown by [127] that a confocal PSF is well modelled by a radially symmetric Gaussian function as:

$$h_{CLSM}(r,z) = exp(-r^2/(2\sigma_r^2))exp(-z^2/(2\sigma_z^2)) \quad , \tag{1.19}$$

where σ is related to the full-width at half-maximum (FWHM) by FWHM $= 2\sqrt{2\ln 2}\sigma$. Both σ_r and σ_z can be estimated from the detected confocal image by using intensity profiles of sub-resolved structures into the image, like unspecifically bound single antibodies or nanosize sub-cellular compartments, together with Gaussian fits. Similarly, it has been shown that the PSF of a STED microscope operating with continuous-wave (CW) lasers (also called CW-STED microscope) is well modeled by [116, 114]:

$$h_{CW-STED}(r,z) = h_{CLSM}(r,z)1/(1 + 4\psi^2 r^2 \varsigma) \quad , \tag{1.20}$$

where: ψ is a constant that depends on the shape of the doughnut-like STED intensity distribution at the focus [67]; ς is the so called saturation factor, which is defined as $\varsigma = I_{STED}/I_s$, I_{STED} being the maximum value of the STED intensity distribution at the focus and I_s being the effective saturation intensity, which can be defined as the intensity at which the probability of fluorescence emission is reduced by half. In the most general case, I_s is a function of the orientation distribution and rotational behavior of the fluorescent marker, as well as of the wavelength, temporal structure and polarization of the inhibition light [67, 115]. An estimation of ψ can be obtained by using scattering images of single isolated 80-nm gold bead. Importantly, $\varsigma = 0$ and $h_{CW-STED} = h_{CLSM}$ if the STED beam intensity is null. Thereby, by taking advantage of having the CW-STED and confocal images of the very same specimen, σ values are estimated as described above. Next, given σ and ψ, the saturation factor ς was estimated using equation (1.20) and the intensity profiles through sub-resolved structures in the CW-STED image. Finally, it has to be mentioned that the Gaussian-based models for the PSF can fail

in the case of thick specimen. In this case images are affected by a depth-variant blur due to spherical aberrations induced by refractive index mismatch between the different media composing the system as well as the specimen [100]. Nevertheless, in practice it is difficult to obtain such a PSF, in spite of the existence of theoretical models accounting for spherical aberrations, because these models depend on some unknown acquisition parameter, such as the refractive index of the specimen. Therefore one needs blind or semi-blind restoration algorithms (see, for instance [64], where an alternating minimization scheme is used in conjunction with SGP as minimization algorithm for depth-variant image deconvolution in confocal microscopy).

1.5 Boundary Effects

In both cases analyzed before, we have to deal with boundary effects. As a consequence of the limited field of view (FOV) of a telescope, it may happen that an extended object is not completely contained within the image domain; in other words the boundaries of the image do not correspond to free sky. Similarly, when is observed a particular of a specimen in a microscope, it is surrounded by other elements and its boundaries are not free (Figure 1.4). Due to the fact that the most efficient RL method is implemented by means of FFT to apply the convolution operator, in these cases it can not be successfully used. Indeed the use of FFT implicitly assumes a periodic continuation of the image outside of the original domain; as a consequence discontinuities generates Gibbs oscillations, also known as ripples, which can propagate inside the image domain and degrade completely the quality of the reconstruction (Figure 1.5).

The basic point is that, as a consequence of the finite extent of the PSF, values of the object outside the FOV significantly contribute to the values of the image in the pixels which are close to the boundary. Therefore one can formulate the problem by introducing these values as unknown parameters which must be estimated by the deconvolution method. The main difficulties generated by this approach are consequently the under-determination of the problem because the number of unknown values is greater than the number of data, and the lost of the computational efficiency given by the FFT, because it cannot be stated as a standard deconvolution problem.

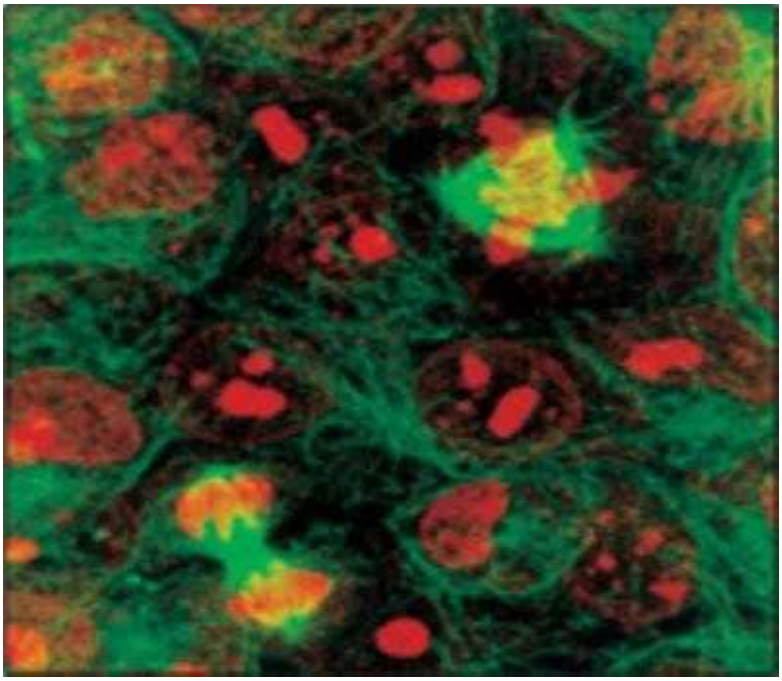


Figure 1.4: Blurred and noisy image of a beta II Tubulin A Antibody.

The purpose of the research made is to extend the application of the SGP method to the problem of multiple image deconvolution and, mainly, to the problem of boundary effect correction. The first problem is basic, for instance, for the reconstruction of the images of the future interferometer of the Large Binocular Telescope (LBT), denoted LINC-NIRVANA [71], while the second problem is important both in single and multiple image deconvolution, as stated before.

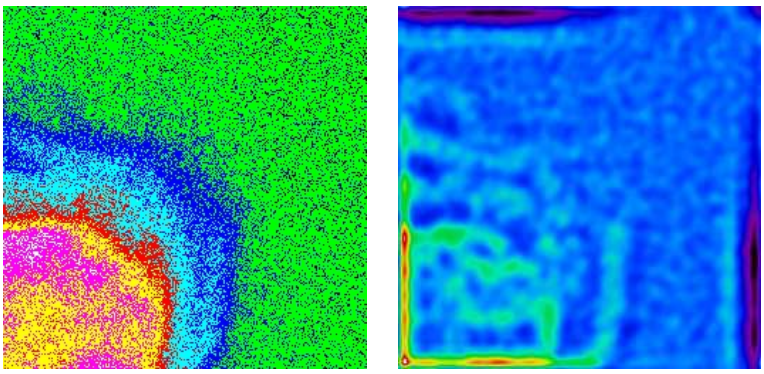


Figure 1.5: On the left side, the blurred and noisy image of an extended galaxy. On the right the reconstruction provided without taking into account boundary effect correction. Ripples are evident on the borders and, with their intensity, they corrupt the entire reconstruction.

Chapter 2

Scaled Gradient Projection (SGP) methods

2.1 The deconvolution method

In this Section is described the monotone SGP algorithm for the minimization of a convex and differentiable function on the non-negative orthant. For the general version, including flux constraint, please refer to Bonettini et al. [23].

The main feature of the gradient projection method introduced here consists in the combination of non-expensive diagonally scaled gradient directions with steplength selection rules specially designed for these directions. Moreover, global convergence properties are ensured by exploiting a non-monotone line-search strategy along the feasible direction [20, 61].

Scaled gradient directions are also used by other popular algorithms for image restoration; see, for example, the projected Newton methods described in [7, 81, 83, 119]. However, these schemes are substantially different from our approach since they require inner linear solvers to compute the non-diagonally scaled gradient direction, do not consider steplength selection strategies and use line-search along the projection arc instead of along the feasible direction [18, p.226]. Other scaled-gradient-based methods are presented in [58, 65]: these approaches do not need a projection step: in fact, thanks to the careful choice of the scaling matrix, each iterate lies in the interior of the feasible set.

The SGP scheme is a gradient method for the solution of the problem (1.2)

ALGORITHM 2.1 Scaled Gradient Projection (SGP) method

Initialization.

Choose the starting point $\boldsymbol{x}^{(0)} \in \Omega$, set the parameters $\beta, \theta \in (0,1)$, $0 < \alpha_{\min} < \alpha_{\max}$ and fix a positive integer M.

Main loop.

FOR $k = 0, 1, 2, \ldots$ do the following steps:

1 Choose the parameter $\alpha_k \in [\alpha_{\min}, \alpha_{\max}]$ and the scaling matrix $D_k \in \mathcal{C}$;

2 Projection: $\tilde{\boldsymbol{x}}^{(k)} = \mathbb{P}_{\Omega, D_k^{-1}}\big(\boldsymbol{x}^{(k)} - \alpha_k D_k \nabla f(\boldsymbol{x}^{(k)})\big)$;
IF $\tilde{\boldsymbol{x}}^{(k)} = \boldsymbol{x}^{(k)}$ THEN stop: $\boldsymbol{x}^{(k)}$ is a stationary point; ENDIF

3 Descent direction: $\boldsymbol{d}^{(k)} = \tilde{\boldsymbol{x}}^{(k)} - \boldsymbol{x}^{(k)}$;

4 Set $\lambda_k = 1$ and $f_{\max} = \max_{0 \le j \le \min(k, M-1)} f(\boldsymbol{x}^{(k-j)})$;

5 Backtracking loop:
IF $f\big(\boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{d}^{(k)}\big) \le f_{\max} + \beta \lambda_k \nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{d}^{(k)}$ THEN
go to Step 6;
ELSE
set $\lambda_k = \theta \lambda_k$ and go to Step 5;
ENDIF

6 Set $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{d}^{(k)}$.

END

Each SGP iteration is based on the descent direction $\boldsymbol{d}^{(k)} = \tilde{\boldsymbol{x}}^{(k)} - \boldsymbol{x}^{(k)}$, where

$$\tilde{\boldsymbol{x}}^{(k)} = \mathbb{P}_{\Omega, D_k^{-1}}\big(\boldsymbol{x}^{(k)} - \alpha_k D_k \nabla f(\boldsymbol{x}^{(k)})\big); \tag{2.1}$$

is defined by combining a scaled steepest descent direction with a projection on the non-negative orthant. The matrix D_k in (2.1) is chosen in the set $\mathcal{D}$ of the $N \times N$ diagonal positive definite matrices whose diagonal elements are bounded between L_1 and L_2, for given thresholds $0 < L_1 < L_2$.

The main SGP steps are given in Algorithm 2.1. The global convergence of the algorithm is obtained by means of the standard monotone Armijo rule in the line-search procedure described in the Step 5 (see Bonettini et al. [23]).

It is worth to take in account that any choice of the steplength $\alpha_k \in [\alpha_{min}, \alpha_{max}]$ and of the scaling matrix $D_k \in \mathcal{D}$ are allowed; this freedom of choice can be therefore successfully exploited for introducing performance improvements and is

described in the following Sections.

Now, before to state the convergence analysis of the SGP algorithm and some strategies for the choice of the steplength and scaling matrix updating, we recall some notations and basic properties.

2.2 Notations and basic properties

Throughout the explanation, the 2-norm of vectors and matrices is denoted by $\|\cdot\|$ while $\|\cdot\|_D$ indicates the vector norm associated to a symmetric positive definite matrix D: $\|\boldsymbol{x}\|_D = \sqrt{\boldsymbol{x}^T D \boldsymbol{x}}$.

Let $\Omega \subset \mathbb{R}^N$ be a closed convex set and D be a symmetric positive definite $N \times N$ matrix, we define the projection operator $\mathbb{P}_{\Omega,D} : \mathbb{R}^N \to \Omega$ as

$$\mathbb{P}_{\Omega,D}(\boldsymbol{x}) \equiv \arg\min_{\boldsymbol{y}\in\Omega} \|\boldsymbol{y}-\boldsymbol{x}\|_D = \arg\min_{\boldsymbol{y}\in\Omega} \left(\phi(\boldsymbol{y}) \equiv \frac{1}{2}\boldsymbol{y}^T D\boldsymbol{y} - \boldsymbol{y}^T D\boldsymbol{x}\right). \tag{2.2}$$

In this case $\boldsymbol{y}$ is not the observed image but just another point inside the domain. We observe that, given the set Ω and the point $\boldsymbol{x}$, the operator $\mathbb{P}_{\Omega,D}(\boldsymbol{x})$ is a continuous function with respect to the elements of the matrix D. From the definition of stationary point and the strict convexity of the function ϕ introduced in (2.2), we have that $\mathbb{P}_{\Omega,D}(\boldsymbol{x})$ is defined also by

$$\left(\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \boldsymbol{x}\right)^T D \left(\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \boldsymbol{y}\right) \leq 0, \qquad \forall\, \boldsymbol{y} \in \Omega. \tag{2.3}$$

This can be proved by evaluating the gradient of the function ϕ in the point $\mathbb{P}_{\Omega,D}(\boldsymbol{x})$

$$\nabla\phi\left(\mathbb{P}_{\Omega,D}(\boldsymbol{x})\right) = D\left(\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \boldsymbol{x}\right)$$

Since by definition $\mathbb{P}_{\Omega,D}(\boldsymbol{x})$ is a constrained stationary point for the problem (2.2), we have:

$$-\nabla\phi(\mathbb{P}_{\Omega,D}(\boldsymbol{x}))^T(\boldsymbol{y} - \mathbb{P}_{\Omega,D}(\boldsymbol{x})) \leq 0, \qquad \forall\, \boldsymbol{y} \in \Omega\ ,$$

and from the symmetry of D and the previously evaluated gradient:

$$(\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \boldsymbol{x})^T D(\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \boldsymbol{y}) \leq 0, \qquad \forall\, \boldsymbol{y} \in \Omega\ .$$

Let $\mathcal{D}_L \subset \mathbb{R}^{N\times N}$ be the compact set of the symmetric positive definite $N \times N$

matrices such that $\|D\| \leq L$ and $\|D^{-1}\| \leq L$, for a given threshold $L > 1$. The next two lemmas state some properties of the projection operator defined in (2.2).

Lemma 2.1 *If $D \in \mathcal{D}_L$, then*

$$\|\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \mathbb{P}_{\Omega,D}(\boldsymbol{z})\| \leq L^2\|\boldsymbol{x} - \boldsymbol{z}\| \tag{2.4}$$

for any $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^N$.

Proof. By applying the condition (2.3) we get

$$\begin{aligned}(\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \boldsymbol{x})^T D(\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \mathbb{P}_{\Omega,D}(\boldsymbol{z})) &\leq 0\\ (\mathbb{P}_{\Omega,D}(\boldsymbol{z}) - \boldsymbol{z})^T D(\mathbb{P}_{\Omega,D}(\boldsymbol{z}) - \mathbb{P}_{\Omega,D}(\boldsymbol{x})) &\leq 0\end{aligned}$$

and, by adding the two inequalities,

$$((\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \boldsymbol{x}) - (\mathbb{P}_{\Omega,D}(\boldsymbol{z}) - \boldsymbol{z}))^T D(\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \mathbb{P}_{\Omega,D}(\boldsymbol{z})) \leq 0,$$

that is

$$\|\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \mathbb{P}_{\Omega,D}(\boldsymbol{z})\|_D^2 \leq (\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \mathbb{P}_{\Omega,D}(\boldsymbol{z}))^T D(\boldsymbol{x} - \boldsymbol{z}). \tag{2.5}$$

If σ_{min} denotes the minimum eigenvalue of the matrix D, for the left hand side of the previous inequality we have

$$\begin{aligned}\|\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \mathbb{P}_{\Omega,D}(\boldsymbol{z})\|_D^2 &\geq \sigma_{min}\|\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \mathbb{P}_{\Omega,D}(\boldsymbol{z})\|^2\\ &= \tfrac{1}{\|D^{-1}\|}\|\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \mathbb{P}_{\Omega,D}(\boldsymbol{z})\|^2\\ &\geq \tfrac{1}{L}\|\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \mathbb{P}_{\Omega,D}(\boldsymbol{z})\|^2\end{aligned}$$

and from (2.5) we obtain

$$\begin{aligned}\tfrac{1}{L}\|\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \mathbb{P}_{\Omega,D}(\boldsymbol{z})\|^2 &\leq \|\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \mathbb{P}_{\Omega,D}(\boldsymbol{z})\|_D^2\\ &\leq (\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \mathbb{P}_{\Omega,D}(\boldsymbol{z}))^T D(\boldsymbol{x} - \boldsymbol{z})\\ &\leq \|\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \mathbb{P}_{\Omega,D}(\boldsymbol{z})\|\ \|D\|\ \|\boldsymbol{x} - \boldsymbol{z}\|\\ &\leq L\|\mathbb{P}_{\Omega,D}(\boldsymbol{x}) - \mathbb{P}_{\Omega,D}(\boldsymbol{z})\|\ \|\boldsymbol{x} - \boldsymbol{z}\|\end{aligned}$$

which yields (2.4). □

Lemma 2.2 *A vector $\boldsymbol{x}_* \in \Omega$ is a stationary point of the problem (1.2) if and only if $\boldsymbol{x}_* = \mathbb{P}_{\Omega,D^{-1}}(\boldsymbol{x}_* - \alpha D\nabla f(\boldsymbol{x}_*))$ for any positive scalar α and for any symmetric positive definite matrix D.*

Proof. Let $\alpha \in \mathbb{R}^+$ and let D be a symmetric positive definite matrix. Assume that $\boldsymbol{x}_* = \mathbb{P}_{\Omega,D^{-1}}(\boldsymbol{x}_* - \alpha D\nabla f(\boldsymbol{x}_*))$. From (2.3) we obtain

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

that is

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

which, being D symmetric and $\alpha > 0$, implies the stationarity condition

$$\nabla f(\boldsymbol{x}_*)^T(\boldsymbol{x}_* - \boldsymbol{x}) \leq 0, \qquad \forall\, \boldsymbol{x} \in \Omega.$$

Conversely, let us assume that $\boldsymbol{x}_* \in \Omega$ is a stationary point of (1.2), and suppose that $\bar{\boldsymbol{x}} = \mathbb{P}_{\Omega,D^{-1}}(\boldsymbol{x}_* - \alpha D\nabla f(\boldsymbol{x}_*))$, with $\bar{\boldsymbol{x}} \neq \boldsymbol{x}_*$. Then, from (2.3) we can write

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

that is

$$\|\bar{\boldsymbol{x}} - \boldsymbol{x}_*\|^2_{D^{-1}} + \alpha\nabla f(\boldsymbol{x}_*)^T(\bar{\boldsymbol{x}} - \boldsymbol{x}_*) \leq 0.$$

The previous inequality yields

$$\nabla f(\boldsymbol{x}_*)^T(\boldsymbol{x}_* - \bar{\boldsymbol{x}}) \geq \frac{\|\bar{\boldsymbol{x}} - \boldsymbol{x}_*\|^2_{D^{-1}}}{\alpha} > 0,$$

which gives a contradiction with the stationarity assumption on $\boldsymbol{x}_*$. □

The Lemma 2.2 shows the effect of the projection operator $\mathbb{P}_{\Omega,D^{-1}}$ on the points $(\boldsymbol{x}_* - \alpha D\nabla f(\boldsymbol{x}_*))$, $\alpha > 0$, when $\boldsymbol{x}_*$ is a stationary point of (1.2). In the case $\bar{\boldsymbol{x}} \in \Omega$ is a non-stationary point, $\mathbb{P}_{\Omega,D^{-1}}(\bar{\boldsymbol{x}} - \alpha D\nabla f(\bar{\boldsymbol{x}}))$ can be exploited to generate a descent direction for the function f in $\bar{\boldsymbol{x}}$. This idea serves as the basis for the method described in Algorithm 2.1. In particular, given $D_k \in \mathcal{D}_L$ and $\alpha_k \in [\alpha_{min}, \alpha_{max}]$, the SGP algorithm makes use of the following direction:

$$\boldsymbol{d}^{(k)} = \mathbb{P}_{\Omega,D_k^{-1}}(\boldsymbol{x}^{(k)} - \alpha_k D_k\nabla f(\boldsymbol{x}^{(k)})) - \boldsymbol{x}^{(k)}\ . \tag{2.6}$$

The properties of this direction are proved in Lemma 2.3, while the behavior of the sequence $\{\boldsymbol{d}^{(k)}\}$ is inspected in Lemma 2.4.

As concerns the steplength selection, we refer to Section 2.4, where we expose the strategy we adopted. The choice of the scaling matrix takes into account the effective form of the function we are minimizing, as well as some additional properties of the optimization problem that has to be solved. For this reason some hints about the choice of the matrix D_k are exposed in Section 2.5, when special minimization problems will be taken into consideration.

Before to discuss the convergence properties of the method, some considerations about its main steps can be useful. If the projection performed in step 2 returns a vector $\tilde{\boldsymbol{x}}^{(k)}$ equal to $\boldsymbol{x}^{(k)}$, then Lemma 2.2 implies that $\boldsymbol{x}^{(k)}$ is a stationary point and the algorithm stops. When $\tilde{\boldsymbol{x}}^{(k)} \neq \boldsymbol{x}^{(k)}$, it is possible to prove that $\boldsymbol{d}_k$ defined in (2.6) is a descent direction for f in $\boldsymbol{x}^{(k)}$ (see Lemma 2.3) and the backtracking loop in step 5 terminates with a finite number of runs; thus the algorithm is well defined.

The nonmonotone line-search strategy implemented in step 5 ensures that $f(\boldsymbol{x}^{(k+1)})$ is lower than the maximum of the objective function on the last M iterations [61]; of course, if $M = 1$ then the strategy reduces to the standard monotone Armijo rule [18].

2.3 A convergence analysis for SGP

In this Section we will focus on the case in which the algorithm generates an infinite sequence of iterates, denoted by $\{\boldsymbol{x}^{(k)}\}$. The main SGP convergence result is stated in Theorem 2.1, whose proof is based on some crucial properties that we report in the next lemmas.

The first two lemmas are concerned with the descent condition and the boundedness of the directions $\boldsymbol{d}^{(k)}$, respectively.

Lemma 2.3 *Assume that $\boldsymbol{d}^{(k)} \neq \boldsymbol{0}$. Then, $\boldsymbol{d}^{(k)}$ is a descent direction for the function f at $\boldsymbol{x}^{(k)}$, that is, $\nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{d}^{(k)} < 0$.*

Proof. If we write the inequality (2.3) with $\boldsymbol{x} = \boldsymbol{x}^{(k)} - \alpha_k D_k \nabla f(\boldsymbol{x}^{(k)})$, $D = D_k^{-1}$

and $\boldsymbol{y} = \boldsymbol{x}^{(k)}$ we have:

$$\left(\mathbb{P}_{\Omega,D_k^{-1}}\left(\boldsymbol{x}^{(k)} - \alpha_k D_k \nabla f(\boldsymbol{x}^{(k)})\right) - \boldsymbol{x}^{(k)} + \alpha_k D_k \nabla f(\boldsymbol{x}^{(k)})\right)^T D_k^{-1} \left(\mathbb{P}_{\Omega,D_k^{-1}}\left(\boldsymbol{x}^{(k)} - \alpha_k D_k \nabla f(\boldsymbol{x}^{(k)})\right) - \boldsymbol{x}^{(k)}\right) \leq 0.$$

By using the definition of the SGP direction (2.6), it follows that:

$$(\boldsymbol{d}^{(k)} + \alpha_k D_k \nabla f(\boldsymbol{x}^{(k)}))^T D_k^{-1} \boldsymbol{d}^{(k)} \leq 0,$$

and then

$$\nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{d}^{(k)} \leq -\frac{{\boldsymbol{d}^{(k)}}^T D_k^{-1} \boldsymbol{d}^{(k)}}{\alpha_k} < 0. \tag{2.7}$$

□

Lemma 2.4 *If the sequence $\{\boldsymbol{x}^{(k)}\}$ is bounded, then also the sequence $\{\boldsymbol{d}^{(k)}\}$ is bounded.*

Proof. From the definition of $\boldsymbol{d}^{(k)}$ in (2.6) and (2.4) we have that, for any k,

$$\begin{aligned} \|\boldsymbol{d}^{(k)}\| &= \|\mathbb{P}_{\Omega,D_k^{-1}}(\boldsymbol{x}^{(k)} - \alpha_k D_k \nabla f(\boldsymbol{x}^{(k)})) - \boldsymbol{x}^{(k)}\| \\ &= \|\mathbb{P}_{\Omega,D_k^{-1}}(\boldsymbol{x}^{(k)} - \alpha_k D_k \nabla f(\boldsymbol{x}^{(k)})) - \mathbb{P}_{\Omega,D_k^{-1}}(\boldsymbol{x}^{(k)})\| \\ &\leq L^2 \|\alpha_k D_k \nabla f(\boldsymbol{x}^{(k)})\| \leq \alpha_{max} L^3 \|\nabla f(\boldsymbol{x}^{(k)})\|. \end{aligned}$$

Let $\bar{\Omega} \subset \Omega$ be a closed and bounded set that contains the iterates $\boldsymbol{x}^{(k)}$. Since ∇f is a continuous function on Ω, then it is bounded in $\bar{\Omega}$ and thus $\{\boldsymbol{d}^{(k)}\}$ is bounded. □

In the next lemmas some properties of the accumulation points of the sequence $\{\boldsymbol{x}^{(k)}\}$ generated by SGP are proved.

Lemma 2.5 *Assume that the subsequence $\{\boldsymbol{x}^{(k)}\}_{k\in K}$, $K \subset \mathbb{N}$, is converging to a point $\boldsymbol{x}_* \in \Omega$. Then, $\boldsymbol{x}_*$ is a stationary point of (1.2) if and only if*

$$\lim_{k\in K} \nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{d}^{(k)} = 0.$$

Proof. Let $\boldsymbol{x}_*$ be a stationary point of (1.2); this means that $\nabla f(\boldsymbol{x}_*)^T \boldsymbol{d} \geq 0$ for

any vector $\boldsymbol{d}$ such that $\boldsymbol{x}_* + \boldsymbol{d} \in \Omega$. Suppose that $\{\nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{d}^{(k)}\}$ does not tend to 0 for $k \in K$. In this case, taking into account Lemma 2.3, we know that there exists $\varepsilon > 0$ and an infinite set $K_1 \subset K$ such that

$$\nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{d}^{(k)} \leq -\varepsilon < 0, \qquad \forall\ k \in K_1.$$

By the compactness of the interval $[\alpha_{min}, \alpha_{max}]$ and of the set $\mathcal{D}_L$, we can extract a set of indices $K_2 \subset K_1$ such that $\alpha_k \to \alpha_*$, $\alpha_* \in [\alpha_{min}, \alpha_{max}]$, and $D_k \to D_*$, $D_* \in \mathcal{D}_L$, for $k \in K_2$; hence, by continuity, we can write $\lim_{k \in K_2} \boldsymbol{d}^{(k)} = \boldsymbol{d}_*$, where

$$\boldsymbol{d}_* = \mathbb{P}_{\Omega, D_*^{-1}}(\boldsymbol{x}_* - \alpha_* D_* \nabla f(\boldsymbol{x}_*)) - \boldsymbol{x}_*. \tag{2.8}$$

Thus,

$$\lim_{k \in K_2} \nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{d}^{(k)} = \nabla f(\boldsymbol{x}_*)^T \boldsymbol{d}_* \leq -\varepsilon < 0. \tag{2.9}$$

Since from definition (2.8) we have that $\boldsymbol{x}_* + \boldsymbol{d}_*$ belongs to Ω, the inequality (2.9) contradicts the stationarity assumption on $\boldsymbol{x}_*$.
On the other hand, let us assume that $\lim_{k \in K} \nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{d}^{(k)} = 0$. Suppose by contradiction that $\boldsymbol{x}_*$ is not a stationary point. Let $K_3 \subset K$ be a set of indices such that $\alpha_k \to \alpha_*$ and $D_k \to D_*$, when k diverges, $k \in K_3$; we have $\lim_{k \in K_3} \boldsymbol{d}^{(k)} = (\mathbb{P}_{\Omega, D_*^{-1}}(\boldsymbol{x}_* - \alpha_* D_* \nabla f(\boldsymbol{x}_*)) - \boldsymbol{x}_*)$. Furthermore, from Lemma 2.2 there exists $\delta > 0$ such that $\|\mathbb{P}_{\Omega, D_*^{-1}}(\boldsymbol{x}_* - \alpha_* D_* \nabla f(\boldsymbol{x}_*)) - \boldsymbol{x}_*\|^2 = \delta$. By exploiting (2.7), we can write, for a sufficiently large $\bar{k} \in K_3$,

$$\nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{d}^{(k)} \leq -\frac{\boldsymbol{d}^{(k)^T} D_k^{-1} \boldsymbol{d}^{(k)}}{\alpha_k} \leq -\frac{\delta}{2\alpha_{max} L} < 0, \qquad \forall\ k \geq \bar{k},\ \ k \in K_3.$$

This contradicts the assumption $\lim_{k \in K} \nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{d}^{(k)} = 0$ and then $\boldsymbol{x}_*$ must be a stationary point. □

Lemma 2.6 *Let $\boldsymbol{x}_* \in \Omega$ be an accumulation point of the sequence $\{\boldsymbol{x}^{(k)}\}$ such that $\lim_{k \in K} \boldsymbol{x}^{(k)} = \boldsymbol{x}_*$, for some $K \subset \mathbb{N}$. If $\boldsymbol{x}_*$ is a stationary point of (1.2), then $\boldsymbol{x}_*$ is an accumulation point also for the sequence $\{\boldsymbol{x}^{(k+r)}\}_{k \in K}$ for any $r \in \mathbb{N}$. Furthermore,*

$$\lim_{k \in K} \|\boldsymbol{d}^{(k+r)}\| = 0, \qquad \forall\ r \in \mathbb{N}.$$

Proof. From Lemma 2.5 we have that

$$\lim_{k\in K} \nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{d}^{(k)} = 0$$

and, from (2.7), we obtain that

$$\lim_{k\in K} \|\boldsymbol{d}^{(k)}\| = 0 \ .$$

Thus

$$\lim_{k\in K} \|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\| = 0 \ ,$$

and this implies that $\boldsymbol{x}_*$ is an accumulation point also for the sequence $\{\boldsymbol{x}^{(k+1)}\}_{k\in K}$. Recalling again Lemma 2.5, we obtain that

$$\lim_{k\in K} \nabla f(\boldsymbol{x}^{(k+1)})^T \boldsymbol{d}^{(k+1)} = 0 \ ;$$

for the same reasons as before, we conclude that

$$\lim_{k\in K} \|\boldsymbol{d}^{(k+1)}\| = 0 \ .$$

Hence, the statement of the lemma follows by induction. □

At this point we may state a convergence result for SGP.

Theorem 2.1 *Assume that the level set* $\Omega_0 = \{\boldsymbol{x} \in \Omega : f(\boldsymbol{x}) \leq f(\boldsymbol{x}^{(0)})\}$ *is bounded. Every accumulation point of the sequence* $\{\boldsymbol{x}^{(k)}\}$ *generated by the SGP algorithm is a stationary point of* (1.2).

Proof. Since every iterate $\boldsymbol{x}^{(k)}$ lies in Ω_0, the sequence $\{\boldsymbol{x}^{(k)}\}$ is bounded and has at least one accumulation point. Let $\boldsymbol{x}_* \in \Omega$ be such that $\lim_{k\in K} \boldsymbol{x}^{(k)} = \boldsymbol{x}_*$ for a set of indices $K \subset \mathbb{N}$. Let us consider separately the two cases

a. $\inf_{k\in K} \lambda_k = 0$;

b. $\inf_{k\in K} \lambda_k = \rho > 0$.

Case a.
Let $K_1 \subset K$ be a set of indices such that $\lim_{k\in K_1} \lambda_k = 0$. This implies that, for $k \in K_1$, k sufficiently large, the backtracking rule fails to be satisfied at least once.

Thus, at the penultimate step of the backtracking loop, we have

$$f(\boldsymbol{x}^{(k)} + \frac{\lambda_k}{\theta}\boldsymbol{d}^{(k)}) > f(\boldsymbol{x}^{(k)}) + \beta\frac{\lambda_k}{\theta}\nabla f(\boldsymbol{x}^{(k)})^T\boldsymbol{d}^{(k)},$$

hence

$$\frac{f(\boldsymbol{x}^{(k)} + \frac{\lambda_k}{\theta}\boldsymbol{d}^{(k)}) - f(\boldsymbol{x}^{(k)})}{\frac{\lambda_k}{\theta}} > \beta\nabla f(\boldsymbol{x}^{(k)})^T\boldsymbol{d}^{(k)}. \tag{2.10}$$

By the mean value theorem, we have that there exits a scalar $t_k \in [0, \frac{\lambda_k}{\theta}]$ such that the left hand side of (2.10) is equal to $\nabla f(\boldsymbol{x}^{(k)} + t_k\boldsymbol{d}^{(k)})^T\boldsymbol{d}^{(k)}$. Thus, the inequality (2.10) becomes

$$\nabla f(\boldsymbol{x}^{(k)} + t_k\boldsymbol{d}^{(k)})^T\boldsymbol{d}^{(k)} > \beta\nabla f(\boldsymbol{x}^{(k)})^T\boldsymbol{d}^{(k)}. \tag{2.11}$$

Since α_k and D_k are bounded, it is possible to find a set of indices $K_2 \subset K_1$ such that $\lim_{k\in K_2}\alpha_k = \alpha_*$ and $\lim_{k\in K_2} D_k = D_*$. Thus the sequence $\{\boldsymbol{d}^{(k)}\}_{k\in K_2}$ converges to the vector $\boldsymbol{d}_* = (\mathbb{P}_{\Omega,D_*^{-1}}(\boldsymbol{x}_* - \alpha_* D_*\nabla f(\boldsymbol{x}_*)) - \boldsymbol{x}_*)$ and, furthermore, $t_k\boldsymbol{d}^{(k)} \to 0$ when k diverges, $k \in K_2$. Taking limits in (2.11) as $k \to \infty$, $k \in K_2$, we obtain

$$(1-\beta)\nabla f(\boldsymbol{x}_*)^T\boldsymbol{d}_* \geq 0.$$

Since $(1-\beta) > 0$ and $\nabla f(\boldsymbol{x}^{(k)})^T\boldsymbol{d}^{(k)} < 0$ for all k, then we necessarily have $\lim_{k\in K_2}\nabla f(\boldsymbol{x}^{(k)})^T\boldsymbol{d}^{(k)} = \nabla f(\boldsymbol{x}_*)^T\boldsymbol{d}_* = 0$. Then, by Lemma 2.5, we conclude that $\boldsymbol{x}_*$ is a stationary point.

Case b.

Let us define the point $\boldsymbol{x}^{(\ell(k))}$ as the point such that

$$f(\boldsymbol{x}^{(\ell(k))}) = f_{max} = \max_{0\leq j\leq \min(k,M-1)} f(\boldsymbol{x}^{(k-j)}).$$

Then, for $k > M-1, k \in \mathbb{N}$, the following condition holds:

$$f(\boldsymbol{x}^{(\ell(k))}) \leq f(\boldsymbol{x}^{(\ell(\ell(k)-1))}) + \beta\lambda_{\ell(k)-1}\nabla f(\boldsymbol{x}^{(\ell(k)-1)})^T\boldsymbol{d}^{(\ell(k)-1)}. \tag{2.12}$$

Since the iterates $\boldsymbol{x}^{(k)}$, $k \in \mathbb{N}$ belong to a bounded set, the monotone non-increasing sequence $\{f(\boldsymbol{x}^{(\ell(k))})\}$ admits a finite limit $\mathcal{L} \in \mathbb{R}$ for $k \in K$. Let $K_3 \subset K$ be a set of indices such that $\lim_{k\in K_3}\lambda_{\ell(k)-1} = \rho_1 \geq \rho > 0$ and $\lim_{k\in K_3}\nabla f(\boldsymbol{x}^{(\ell(k)-1)})^T\boldsymbol{d}^{(\ell(k)-1)}$ exists (recall that, from Lemma 2.4, the sequence $\{\boldsymbol{d}^{(k)}\}_{k\in\mathbb{N}}$ is bounded); taking

limits on (2.12) for $k \in K_3$ we obtain

$$\mathcal{L} \leq \mathcal{L} + \beta\rho_1 \lim_{k\in K_3} \nabla f(\boldsymbol{x}^{(\ell(k)-1)})^T \boldsymbol{d}^{(\ell(k)-1)},$$

that is

$$\lim_{k\in K_3} \nabla f(\boldsymbol{x}^{(\ell(k)-1)})^T \boldsymbol{d}^{(\ell(k)-1)} \geq 0.$$

Recalling that $\nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{d}^{(k)} < 0$ for any k, the previous inequality implies that

$$\lim_{k\in K_3} \nabla f(\boldsymbol{x}^{(\ell(k)-1)})^T \boldsymbol{d}^{(\ell(k)-1)} = 0. \tag{2.13}$$

Then, by the Lemma 2.5, (2.13) implies that every accumulation point of the sequence $\{\boldsymbol{x}^{(\ell(k)-1)}\}_{k\in K_3}$ is a stationary point of (1.2).
Let us prove that the point $\boldsymbol{x}_*$ is an accumulation point of $\{\boldsymbol{x}^{(\ell(k)-1)}\}_{k\in K_3}$.
The definition of $\boldsymbol{x}^{(\ell(k))}$ implies that $k - M + 1 \leq \ell(k) \leq k$. Thus we can write

$$\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(\ell(k)-1)}\| \leq \sum_{j=0}^{k-\ell(k)} \lambda_{\ell(k)-1+j} \, \|\boldsymbol{d}^{(\ell(k)-1+j)}\|, \qquad k \in K. \tag{2.14}$$

Let $K_4 \subset K_3$ be a subset of indices such that the sequence $\{\boldsymbol{x}^{(\ell(k)-1)}\}_{k\in K_4}$ converges to an accumulation point $\bar{\boldsymbol{x}} \in \Omega$. Recalling that, from (2.13) and Lemma 2.5, $\bar{\boldsymbol{x}}$ is a stationary point of (1.2), we can apply Lemma 2.6 to obtain that $\lim_{k\in K_4} \|\boldsymbol{d}^{(\ell(k)-1+j)}\| = 0$ for any $j \in \mathbb{N}$. By using (2.14) we conclude that

$$\lim_{k\in K_4} \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(\ell(k)-1)}\| = 0. \tag{2.15}$$

Since

$$\|\boldsymbol{x}_* - \boldsymbol{x}^{(\ell(k)-1)}\| \leq \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(\ell(k)-1)}\| + \|\boldsymbol{x}^{(k)} - \boldsymbol{x}_*\|$$

and $\lim_{k\in K} \boldsymbol{x}^{(k)} = \boldsymbol{x}_*$, then (2.15) implies that $\boldsymbol{x}_*$ is an accumulation point also for the sequence $\{\boldsymbol{x}^{(\ell(k)-1)}\}_{k\in K_3}$. Hence, we conclude that $\boldsymbol{x}_*$ is a stationary point of (1.2). □

2.4 Update the steplength parameter

Steplength selection rules in gradient methods have received an increasing interest in the last years from both the theoretical and the practical point of view. Following the original ideas of Barzilai and Borwein (BB) [8], we can regard the matrix $B(\alpha_k)$ as an approximation of the Hessian $\nabla^2 f(\boldsymbol{x}^{(k)})$ and derive two updating rules for α_k by forcing quasi-Newton properties on $B(\alpha_k)$:

$$\alpha_k^{\mathrm{BB1}} = \arg\min_{\alpha_k \in \mathbb{R}} \|B(\alpha_k)\boldsymbol{s}^{(k-1)} - \boldsymbol{z}^{(k-1)}\| \tag{2.16}$$

and

$$\alpha_k^{\mathrm{BB2}} = \arg\min_{\alpha_k \in \mathbb{R}} \|\boldsymbol{s}^{(k-1)} - B(\alpha_k)^{-1}\boldsymbol{z}^{(k-1)}\| \,, \tag{2.17}$$

where $\boldsymbol{s}^{(k-1)} = \left(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\right)$ and $\boldsymbol{z}^{(k-1)} = \left(\nabla f(\boldsymbol{x}^{(k)}) - \nabla f(\boldsymbol{x}^{(k-1)})\right)$.
In non-scaled gradient methods $B(\alpha_k) = (\alpha_k I)^{-1}$, so the above minimization problems reduce to the standard BB rules:

$$\alpha_k^{\mathrm{BB1}} = \frac{{\boldsymbol{s}^{(k-1)}}^T \boldsymbol{s}^{(k-1)}}{{\boldsymbol{s}^{(k-1)}}^T \boldsymbol{z}^{(k-1)}} \,, \qquad \alpha_k^{\mathrm{BB2}} = \frac{{\boldsymbol{s}^{(k-1)}}^T \boldsymbol{z}^{(k-1)}}{{\boldsymbol{z}^{(k-1)}}^T \boldsymbol{z}^{(k-1)}} \,. \tag{2.18}$$

Several steplength updating strategies have been designed to accelerate the slow convergence exhibited in most cases by standard gradient methods, and a lot of effort has been put into explaining the effects of these strategies [36, 37, 38, 52, 53, 54, 128]. On the other hand, numerical experiments on randomly generated, library and real-life test problems have confirmed the remarkable convergence rate improvements involved by some BB-like steplength selections [37, 38, 122, 53, 105, 126, 128]. Among the alternation strategies proposed in literature, we summarize the approaches introduced in [37], [128] and [53], and introduce another little modification arose from numerical experience.

Cyclic Barzilai–Borwein (CBB) method

The Cyclic Barzilai-Borwein (CBB) method, presented in [37] exploits the same stepsize for a number m of consecutive iterations:

$$\begin{cases} (\alpha_k^{BB1}, & j = 1) & \text{if } k = 1 \text{ or } j = m, \\ (\alpha_{k-1}, & j = j + 1) & \text{otherwise,} \end{cases} \tag{2.19}$$

As pointed out by the authors, the choice of m has a significant impact on performances. For this reason, in the same paper the Adaptive Cyclic Barzilai-Borwein (ACBB) method is proposed:

$$\begin{cases} (\alpha_k^{BB1}, & j = 1) & \text{if } k = 1 \text{ or } j = m \text{ or } \nu_k \geq \beta, \\ (\alpha_{k-1}, & j = j + 1) & \text{otherwise.} \end{cases} \tag{2.20}$$

The meaning of the condition used in (2.20) arises from considerations that can be made in the quadratic programming framework. In [35] is stressed that, if a gradient method is used with a constant stepsize infinitely often when minimizing a quadratic form where the Hessian A has no multiple eigenvalues, the gradient $\boldsymbol{g}^{(k)}$ must approximate an eigenvector of A. Then, in quadratic programming, the following definition of ν_k can be used:

$$\nu_k = \frac{{\boldsymbol{g}^{(k)}}^T A \boldsymbol{g}^{(k)}}{\|\boldsymbol{g}^{(k)}\| \|A\boldsymbol{g}^{(k)}\|} \ . \tag{2.21}$$

If $\boldsymbol{g}^{(k)}$ is exactly an eigenvector of A, the value of ν_k is 1; for this reason a value $\beta = 0.95$ is suggested and the formula

$$\nu_k = \frac{{\boldsymbol{s}^{(k-1)}}^T \boldsymbol{z}^{(k-1)}}{\|\boldsymbol{s}^{(k-1)}\|_2 \|\boldsymbol{z}^{(k-1)}\|_2} \tag{2.22}$$

is suggested for the generalization of (2.21) when minimizing non quadratic objective functions.

Adaptive Barzilai-Borwein (ABB) method

In [128] the unconstrained minimization of quadratic functionals is taken in consideration. Recalling two steplength choices widely used in gradient methods for quadratic minimization:

$$\alpha_k^{SD} = \arg\min_{\alpha \in \mathbb{R}} f(\boldsymbol{x}^{(k)} - \alpha \boldsymbol{g}^{(k)}) = \frac{{\boldsymbol{g}^{(k)}}^T \boldsymbol{g}^{(k)}}{{\boldsymbol{g}^{(k)}}^T A \boldsymbol{g}^{(k)}} \ , \tag{2.23}$$

$$\alpha_k^{MG} = \arg\min_{\alpha \in \mathbb{R}} \|g(\boldsymbol{x}^{(k)} - \alpha \boldsymbol{g}^{(k)})\| = \frac{{\boldsymbol{g}^{(k)}}^T A \boldsymbol{g}^{(k)}}{{\boldsymbol{g}^{(k)}}^T A^T A \boldsymbol{g}^{(k)}} \ , \tag{2.24}$$

the authors define two gradient methods: the Steepest Descent (SD) gradient method that uses (2.23) and the Minimum Gradient (MG) method that exploits (2.24). The worst case behaviour of both the algorithms is illustrated in a two dimensional problem, and the value of the following ratio is derived:

$$\frac{\alpha_k^{MG}}{\alpha_k^{SD}} \ . \tag{2.25}$$

When using the SD method in the worst case behaviour, the above ratio is over 0.5, while, if one is using MG, the quantity in (2.25) approaches its minimum: moreover, it reaches 0 if the problem is severely ill-conditioned. The authors then suggest a simple method that uses both the steplengths in (2.23) and (2.24) by providing the following scheme as decision rule:

$$\alpha_k = \begin{cases} \alpha_k^{MG} & \text{if } \alpha_k^{MG}/\alpha_k^{SD} > \tau \\ \alpha_k^{SD} & \text{otherwise,} \end{cases} \tag{2.26}$$

and suggesting τ close to 0.5.
By considering the updating formula $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{g}^{(k)}$, starting from (2.23) and (2.24) the following relations can be found:

$$\begin{array}{rcl} \alpha_k^{BB1} & = & \alpha_{k-1}^{SD} \ , \\ \alpha_k^{BB2} & = & \alpha_{k-1}^{MG} \ . \end{array} \tag{2.27}$$

In the test problem evaluated by the authors, the sign of $\alpha_k^{MG}/\alpha_k^{SD} - \tau$ is opposite to that of $\alpha_{k-1}^{MG}/\alpha_{k-1}^{SD} - \tau$: for this reason the decision rule is modified as follow:

$$\alpha_k = \begin{cases} \alpha_k^{BB2} & \text{if } \alpha_k^{BB2}/\alpha_k^{BB1} < \tau \\ \alpha_k^{BB1} & \text{otherwise,} \end{cases} \tag{2.28}$$

In [53] a modified version of (2.28) is proposed, called ABBmin1, in order to allow the same step to be reused in some consecutive iterations:

$$\alpha_k = \begin{cases} \min\{\alpha_j^{BB2} \ \mid \ j = \max\{1, k-m\}, \ldots, k\} & \text{if } \alpha_k^{BB2}/\alpha_k^{BB1} < \tau \\ \alpha_k^{BB1} & \text{otherwise,} \end{cases} \tag{2.29}$$

ALGORITHM 2.2 SGP Steplength Selection

IF $k = 0$ THEN

set $\alpha_0 \in [\alpha_{min}, \alpha_{max}]$, $\tau_1 \in (0,1)$ and a non-negative integer M_α;

ELSE

IF $\boldsymbol{s}^{(k-1)^T} D_k^{-1} \boldsymbol{z}^{(k-1)} \leq 0$ THEN

$\alpha_k^{BB1} = \alpha_{max}$;

ELSE

$\alpha_k^{BB1} = \max\left\{\alpha_{min}, \min\left\{\frac{\boldsymbol{s}^{(k-1)^T} D_k^{-1} D_k^{-1} \boldsymbol{s}^{(k-1)}}{\boldsymbol{s}^{(k-1)^T} D_k^{-1} \boldsymbol{z}^{(k-1)}}, \alpha_{max}\right\}\right\}$;

ENDIF

IF $\boldsymbol{s}^{(k-1)^T} D_k \boldsymbol{z}^{(k-1)} \leq 0$ THEN

$\alpha_k^{BB2} = \alpha_{max}$;

ELSE

$\alpha_k^{BB2} = \max\left\{\alpha_{min}, \min\left\{\frac{\boldsymbol{s}^{(k-1)^T} D_k \boldsymbol{z}^{(k-1)}}{\boldsymbol{z}^{(k-1)^T} D_k D_k \boldsymbol{z}^{(k-1)}}, \alpha_{max}\right\}\right\}$;

ENDIF

IF $\alpha_k^{BB2} / \alpha_k^{BB1} \leq \tau_k$ THEN

$\alpha_k = \min\left\{\alpha_j^{BB2},\ j = \max\{1, k - M_\alpha\}, \ldots, k\right\}$; $\quad \tau_{k+1} = \tau_k * 0.9$;

ELSE

$\alpha_k = \alpha_k^{BB1}$; $\quad \tau_{k+1} = \tau_k * 1.1$;

ENDIF

ENDIF

Steplength selection for the Scaled Gradient approach

First of all we must rewrite, in case of a scaled gradient method, the two BB rules usually exploited by the main steplength updating strategies.
Using $B(\alpha_k) = (\alpha_k D_k)^{-1}$ in (2.16) and (2.17), the steplengths

$$\alpha_k^{BB1} = \frac{\boldsymbol{s}^{(k-1)^T} D_k^{-1} D_k^{-1} \boldsymbol{s}^{(k-1)}}{\boldsymbol{s}^{(k-1)^T} D_k^{-1} \boldsymbol{z}^{(k-1)}} \tag{2.30}$$

and

$$\alpha_k^{BB2} = \frac{\boldsymbol{s}^{(k-1)^T} D_k \boldsymbol{z}^{(k-1)}}{\boldsymbol{z}^{(k-1)^T} D_k D_k \boldsymbol{z}^{(k-1)}} \tag{2.31}$$

are obtained.

At this point, inspired by the steplength alternations previously summarized, we propose an updating rule for SGP which adaptively alternates the values pro-

vided by (2.30) and (2.31). The details of the SGP steplength selection are given in Algorithm 2.2. This rule decides the alternation between two different selection strategies by means of the variable threshold τ_k instead of a constant parameter as done in (2.28) and (2.29). This trick makes the choice of τ_0 less important for the SGP performance and, in our experience, seems able to avoid the drawbacks due to the use of the same steplength rule in too many consecutive iterations. We have to mention that when the scaled versions of the BB rules are used, the inequality $\alpha_k^{(BB2)} \leq \alpha_k^{(BB1)}$ is not always true; nevertheless, a wide computational study suggests that this alternation criterion is still convenient in terms of convergence rate in comparison with the use of a single BB rule (Bonettini et al. [23], Favati et al. [50], Zanella et al. [124]). Furthermore, in our experience, the use of the BB values provided by (2.29) in the first iterations slightly improves the reconstruction accuracy and, consequently, in the proposed SGP version we begin the steplength alternation only after the first 20 iterations.

2.5 Updating the scaling matrix

Concerning the choice of the scaling matrix D_k, it takes into account the special form of the function $f(x)$ we are minimizing and needs to be faced separately for each application considered. In the case of the minimization of the KL divergence we use the scaling suggested by its additive version (1.13), corrected with a threshold assuring that the scaling matrix belongs to $\mathcal{D}$

$$D_k = \operatorname{diag}\left(\min\left\{L_2, \max\left\{L_1, x_i^{(k)}\right\}\right\}\right) \quad . \tag{2.32}$$

Similarly, the analysis of SGM suggests the following scaling in the application of SGP to the minimization of $f_\lambda(x; y)$

$$D_k = \operatorname{diag}\left(\min\left\{L_2, \max\left\{L_1, \frac{x_i^{(k)}}{1 + \lambda(V_1(x^{(k)}))_i}\right\}\right\}\right) \quad . \tag{2.33}$$

where $V_1(x^{(k)})$ is a non-negative array/cube defined by an appropriate splitting of $\nabla f_1(x^{(k)})$. In the case of quadratic (or Tikhonov) regularization, we recall that $V_1(x^{(k)}) = x^{(k)}$.

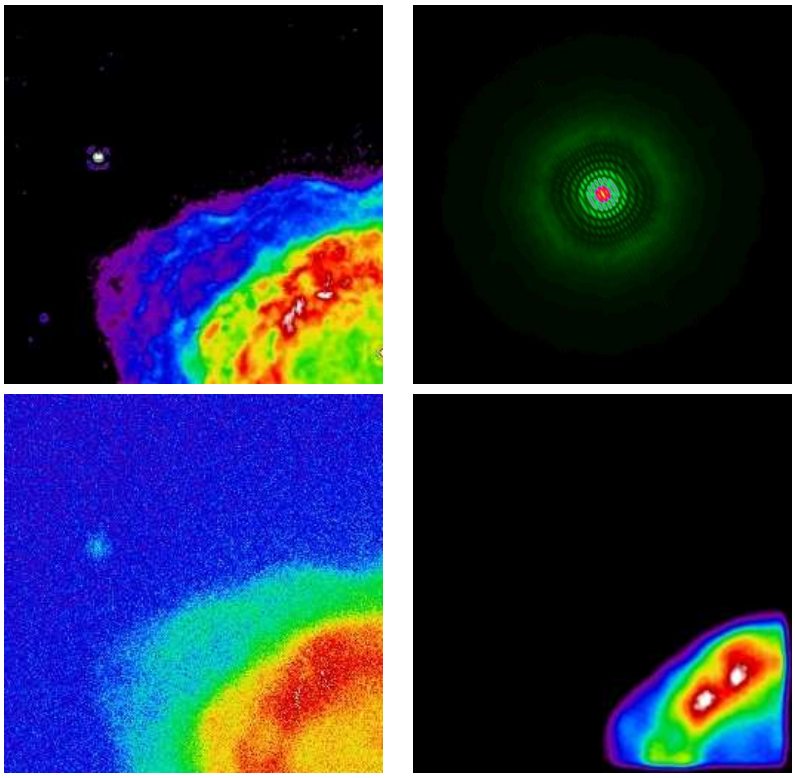


Figure 2.1: In the top left corner we have the clean image, top right the correspondent PSF, bottom left the burred and noisy image, bottom right the reconstructed image with boundary effects.

2.6 Boundary effect correction

If the target $\boldsymbol{x}$ is not completely contained in the image domain, due to the limited field of view of the telescope, then the previous deconvolution methods produce annoying boundary artifacts (see Figure 2.1). To overcome this problem we focus on an approach proposed in Bertero & Boccacci [12] for single image deconvolution and in Anconelli et al. [5, 4] for multiple image deconvolution. Here we give the equations in the case of multiple images; single image corresponds to $p = 1$.

Assuming that the detected image $\boldsymbol{y}$ has size $N = n \times n$, it receives contributes to its pixels by the PSF over a broader domain. This depends on the extension of the PSF and, as an upper bound, stating that the PSF is as broader as the FOV, we have to operate on a domain of size $2n \times 2n$. The idea is to reconstruct the

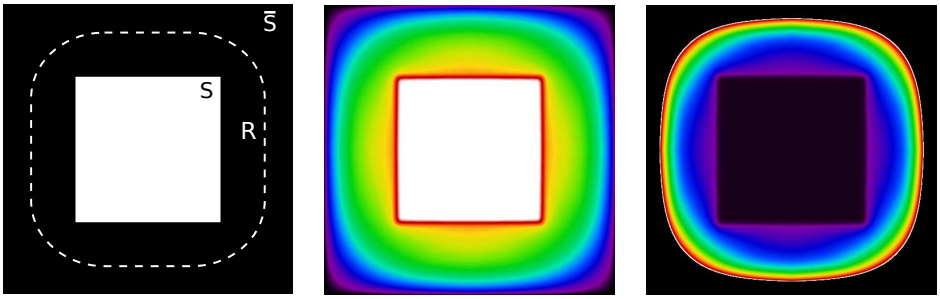


Figure 2.2: Graphical exemplification of the model. On the left the domains S, R and $\bar{S}$, with R considered with the σ threshold. At center the values of $\boldsymbol{\varphi}$ (logarithmic scale) for the proposed PSF. On the right, the values of the modified scaling $\frac{\boldsymbol{M}_R}{\boldsymbol{\varphi}}$. The black area outside the coloured one is considered zero.

object $\boldsymbol{x}$ over a domain broader than that of the detected images and to merge, by zero padding, the arrays of the images and of the object into arrays with a dimension such that their Fourier transform can be computed by means of FFT.

We denote by $\bar{S}$ the set of values of the multi-index labelling the pixels of the broader arrays, by R that of the object array and by S that of the image arrays, so that $S \subset R \subset \bar{S}$. It is obvious that also the PSFs must be defined over $\bar{S}$ and this can be done in different ways, depending on the specific problem one is considering. Also the PSFs must be normalized to unit volume over $\bar{S}$. The reconstruction of $\boldsymbol{x}$ outside S, is not reliable in most cases, but its reconstruction inside S is practically free of boundary artifacts, as shown in the papers cited above and in the experiments of Chapter 5.

If we denote by $\boldsymbol{M}_R$, $\boldsymbol{M}_S$ the arrays, defined over $\bar{S}$, which are 1 over R, S respectively and 0 outside, we define the following matrices A_j and A_j^T:

$$(A_j\boldsymbol{x})(\mathbf{m}) = \boldsymbol{M}_S(\mathbf{m}) \sum_{\mathbf{n}\in\bar{S}} \boldsymbol{K}_j(\mathbf{m}-\mathbf{n})\boldsymbol{M}_R(\mathbf{n})\boldsymbol{x}(\mathbf{n}) \tag{2.34}$$

$$(A_j^T\boldsymbol{y})(\mathbf{n}) = \boldsymbol{M}_R(\mathbf{m}) \sum_{\mathbf{m}\in\bar{S}} \boldsymbol{K}_j(\mathbf{m}-\mathbf{n})\boldsymbol{M}_S(\mathbf{m})\boldsymbol{y}(\mathbf{m}) \ . \tag{2.35}$$

In the second equation $\boldsymbol{y}$ denotes a generic array defined over $\bar{S}$. Both matrices can be easily computed by means of FFT. Then, with these definitions, the data fidelity function is given again by Eq. (1.15), with S replaced by $\bar{S}$, while its

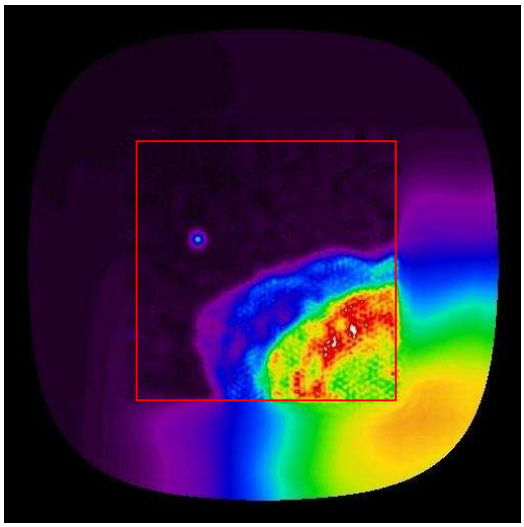


Figure 2.3: The reconstruction of the same object in Figure 2.1 over the broader domain R. The image obtained in the center of the image (red contour) is considered and the pixels in $R \setminus S$ are significant only during the reconstruction process.

gradient is now given by

$$\nabla f_0(\boldsymbol{x};\boldsymbol{y}) = \sum_{j=1}^{p} \left\{ A_j^T \mathbf{1} - A_j^T \frac{\boldsymbol{y}_j}{A_j \boldsymbol{x} + \boldsymbol{b}_j} \right\} \quad , \tag{2.36}$$

suggesting the introduction of the following functions

$$\begin{aligned} \boldsymbol{\varphi}_j(\mathbf{n}) &= (A_j^T \mathbf{1})(\mathbf{n}) \quad , \quad \boldsymbol{n} \in \bar{S} \quad , \\ \boldsymbol{\varphi}(\mathbf{n}) &= \sum_{j=1}^{p} \boldsymbol{\varphi}_j(\mathbf{n}) \quad . \end{aligned} \tag{2.37}$$

These functions can be used for defining the reconstruction domain R, since they can be very small or zero in pixels of $\bar{S}$, depending on the behaviour of the PSFs. Given a threshold value σ, we use the following definition

$$R = \{ \boldsymbol{n} \in \bar{S} \mid \boldsymbol{\varphi}_j(\boldsymbol{n}) \geq \sigma; j = 1, .., p \} \quad . \tag{2.38}$$

Then the RL algorithm, with boundary effect correction, is given by

$$\boldsymbol{x}^{(k+1)} = \frac{\boldsymbol{M}_R}{\boldsymbol{\varphi}} \boldsymbol{x}^{(k)} \sum_{j=1}^{p} A_j^T \frac{\boldsymbol{y}_j}{A_j \boldsymbol{x}^{(k)} + \boldsymbol{b}_j} \quad , \tag{2.39}$$

the quotient being zero in the pixels outside R. Similarly, the OSEM algorithm, with boundary effect correction is given by Algorithm 1.1 with (1.18) replaced by

$$\boldsymbol{h}^{(j)} = \frac{\boldsymbol{M}_R}{\boldsymbol{\varphi}_j} \boldsymbol{h}^{(j-1)} \left(A_j^T \frac{\boldsymbol{y}_j}{A_j \boldsymbol{h}^{(j-1)} + \boldsymbol{b}_j} \right) \quad . \tag{2.40}$$

As far as the SGP algorithm concerns, the boundary effect correction is incorporated in the scaling matrix

$$D_k = \frac{\boldsymbol{M}_R}{\boldsymbol{\varphi}} \operatorname{diag} \left(\min \left\{ L_2, \ \max \left\{ L_1, \boldsymbol{x}^{(k)} \right\} \right\} \right) \tag{2.41}$$

while all the other steps remain unchanged.

Chapter 3

SGP Parallel Implementation

We focus now on the different implementations of the SGP algorithm provided in Section 2.1. As we stated before, in many large-scale applications these improved algorithms do not provide the expected reconstruction in a reasonable time. In these cases, the modern multiprocessor architectures represent an important resource for reducing the reconstruction time. Actually, one can consider different possibilities for a parallel computational scenario.

One is the use of *Graphics Processing Units* (GPUs): they were originally designed to perform many simple operations on matrices and vectors with high efficiency and low accuracy (single precision arithmetic), but they have recently seen a huge development of both computational power and accuracy (double precision arithmetic), while still retaining compactness and low price.

Another possibility is the use of last-generation multi-core CPUs, where general-purpose, very powerful computational cores are integrated inside the same CPU and a bunch of CPUs can be hosted by the same motherboard, sharing a central memory: they can perform completely different and asynchronous tasks, as well as cooperate by suitably distributing the workload of a complex task.

Additional opportunities are offered by the more classical clusters of nodes, usually connected in different distributed-memory topologies to form large-scale high-performance machines with tens to hundred-thousands of processors. Needless to say, various mix of these architectures (such as clusters of GPUs) are also possible and sold, indeed. It should be noticed, however, that all the mentioned scenarios can exist even in very small-sized and cheap configurations. This is particularly relevant for GPUs: initially targeted at 3D graphics applications, they

have been employed in many other scientific computing areas, such as signal and image reconstruction [87, 103].

Recent applications show that in many cases GPU performances are comparable to those of a medium-sized cluster, at a fraction of its cost. Thus, also small laboratories, which cannot afford a cluster, can benefit from a substantial reduction of computing time compared to a standard CPU system. Nevertheless, for very large problems, as 3D imaging in confocal microscopy, the size of GPU's on-devices dedicated memory can become a limit to performance.

For this reason, the ability to exploit the scalability of clusters by means of standard MPI implementations is still crucial for facing very large-scale applications.

3.1 Computational features

The description of the SGP algorithm provided in Section 2.1 show the presence of several ingredients on which the success of the recipe depends: the choice of the starting point, the selection of the parameters defining the method and the stopping criterion. In the following we will briefly describe which choices have been done in the numerical experimentation, together with some comments on the parallel implementation of the algorithm.

3.1.1 Initialization

As far as the SGP initial point $\boldsymbol{x}^{(0)}$ concerns, any non-negative image is allowed. The possible choices implemented in our code are:

- the null image $\boldsymbol{x}^{(0)} = \mathbf{0}$;
- the noisy image $\boldsymbol{y}$ (or, in the case of multiple deconvolution, the noisy image $\boldsymbol{y}_1$ corresponding to the first PSF $\boldsymbol{K}_1$);
- a constant image with pixels value equal to the background-subtracted flux (or mean flux in the case of multiple deconvolution) of the noisy data divided by the number of pixels. If the boundary effect correction is considered, only the pixels in the object array R become equal to this constant, while the remaining values of $\bar{S}$ are set to zero;

- any input image provided by the user.

The constant image $\boldsymbol{x}^{(0)}$ has been chosen for our numerical experiments, which is also the initial point used for RL.

3.1.2 SGP parameters setting

Even if the number of SGP parameters is certainly higher than that of the RL and OSEM approaches, the huge amount of tests carried out in several applications led to an optimization of these values which allows the user to have at his disposal a robust approach without the need of a problem-dependent parameters tuning. Some of these values have been fixed according to the original paper of Bonettini et al. [23], as the line-search parameters β and θ which have been set equal to 10^{-4} and 0.4, respectively. Also most of the steplength parameters remained unchanged, as $\alpha_0 = 1.3$, $\tau_1 = 0.5$, $\alpha_{max} = 10^5$ and $M_\alpha = 3$, while α_{min} has been set equal to 10^{-5}.

The main change concerned the choice of the bounds (L_1, L_2) for the scaling matrices. While in the original paper the choice was a couple of fixed values $(10^{-10}, 10^{10})$, independent of the data, we decided to automatically adapt these bounds to the input image: we perform one step of the RL method and tune the parameters (L_1, L_2) according to the min/max values of the resulting image according to the rule

IF $y_{\max}/y_{\min} < 50$ THEN
  $L_1 = y_{\min}/10$;
  $L_2 = y_{\max} \cdot 10$;
ELSE
  $L_1 = y_{\min}$;
  $L_2 = y_{\max}$;
ENDIF

3.1.3 Stopping rules

As mentioned in Sec 1.2, in many instances both RL and SGP must not be pushed to convergence and early stopping of the iterations is required for obtaining sensible reconstructions. In our code, we have introduced different stopping criteria which can be chosen by the user according to his/her purposes:

- fixed number of iterations. The user can decide how many iterations of SGP must be computed;

- difference between consecutive values of the objective function. In such a case a stopping criterion based on the convergence of the data-fidelity function to its minimum value is introduced. Iteration is stopped when

$$|f_0(\boldsymbol{x}^{(k+1)};\boldsymbol{y}) - f_0(\boldsymbol{x}^{(k)};\boldsymbol{y})| \leq tol\ f_0(\boldsymbol{x}^{(k)};\boldsymbol{y}) \quad , \tag{3.1}$$

  where *tol* is a parameter which can be selected by the user;

- minimization of the reconstruction error. This criterion can be used in a simulation study. If one knows the object $\widetilde{\boldsymbol{x}}$, used for generating the synthetic images, then one can stop the iterations when the relative reconstruction error

$$\rho^{(k)} = \frac{|\boldsymbol{x}^{(k)} - \widetilde{\boldsymbol{x}}|}{|\widetilde{\boldsymbol{x}}|} \tag{3.2}$$

  reaches a minimum error. A very frequently used measure of error is given by the ℓ_2 norm, i.e. $|.| = ||.||_2$ and this is the criterion implemented in our code;

- use of a discrepancy criterion. In the case of real data one can use a given value of some measure of the "distance" between the real data and the data computed by means of the current iteration. A recently proposed criterion consists in defining the following "discrepancy function" for p images $\boldsymbol{y}_j$ with size N

$$\mathcal{D}^{(k)} = \frac{2}{p\ N} f_0(\boldsymbol{x}^{(k)};\boldsymbol{y}) \quad , \tag{3.3}$$

  and stopping the iterations when $\mathcal{D}^{(k)} = b$, where b is a given number close to 1.

The last stopping rule deserves a few comments. In Bertero et al. [17] it is shown that, in the case of a single image, if $\widetilde{\boldsymbol{x}}$ is the object generating the noisy image $\boldsymbol{y}$, then the expected value of $f_0(\widetilde{\boldsymbol{x}};\boldsymbol{y})$ is close to $N/2$. This property is used for selecting a value of the regularization parameter when the image reconstruction problem is formulated as the minimization of the KL divergence with the addition of a suitable regularization term. It is supported by proving that in some important

cases it provides a unique value of the regularization parameter. Moreover, it is also shown that the quantity $\mathcal{D}^{(k)}$, defined in Eq. (3.3), decreases for increasing k, starting from a value greater than 1. Therefore, it can be used for a stopping criterion. Preliminary numerical experiments described in that paper show that it can provide a sensible stopping rule at least in simulation studies.

3.2 GPU and MPI implementations

For what concern the astronomical application, our implementation of the SGP algorithm has been written in IDL (Interactive Data Language), a well known and frequently used language in astronomical environment. This data-analysis programming language is well suited to work with images, using optimized built-in vector operations. Nevertheless is not designed to emphasize performance over usability.

With respect to the microscopy environment, a serial C++ and parallel MPI and CUDA version of the algorithms have been developed. In this case the main target was to obtain a "real-time" deconvolution process, therefore computational efficiency has been preferred over easiness of implementation or higher level languages.

3.2.1 IDL implementation

As already shown in Ruggiero et al [103], the C++ implementation of the SGP algorithm is well suited for parallelization and good computational speedup is obtained exploiting the CUDA technology. As already stated, CUDA is a framework developed by Nvidia that enables the use of GPU (Graphics Processing Unit) for programming. These graphics cards are nowadays in many personal computers and their core is highly parallel, made of several hundreds of computational units. Many recent applications show that the speedup obtained with this technology is significant and its cost is much lower than a medium-size cluster. We remark that memory management is crucial to have best performance when using GPU. The transfer speed of data from central memory to GPU is much slower than the GPU-to-GPU transfer so, for maximizing the GPU benefits, it is very important to reduce the CPU-to-GPU memory communications and keep all the problem data on the GPU memory.

CUDA capability is obtained in IDL by the use of GPUlib, a software library that enables GPU-accelerated calculations, developed by Tech-X Corporation. It has to be noted that the FFT routine included in the current (at the time of developing) version of GPUlib (1.4.4) is available only in single precision. Results from this function are slightly different from the ones obtained in double precision by IDL, bringing some numerical differences in our experiments.

3.2.2 C++ and CUDA

We base our GPU computational study on the Nvidia Graphics adapters, in particular within the manufacturer-provided framework called CUDA (Compute Unified Device Architecture). For further information see `http://www.nvidia.com/cuda`. By means of CUDA, it is possible to program a GPU using a C-like programming language. In this paradigm the CPU controls the instruction flow, the communications with the peripherals and starts the single computing tasks on the GPU.

The GPU, composed by a number of streaming multiprocessors, performs the raw computation tasks, using the problem data stored into the graphics memory. The GPU core is highly parallel: each streaming multiprocessor is composed by many cores (the number depends on the different generation of the architecture, see Table 3.1 and Figure3.1), a high-speed RAM block shared among the cores and a cache. All the streaming multiprocessors can access the main global memory where, typically, the problem data are stored. The idea is to divide the computation among blocks (who can be addressed separately) and for each block obtain a finer parallelization grain dividing computation between the actual cores (threads) as shown in Figure 3.2.

The GPU is usually connected to the CPU with a PCI-Express bus, which grants a 8 GB/sec transfer rate. It should be noted that the transfer speed is much slower than the GPU-to-GPU transfer so, for maximizing the GPU benefits, it is very important to reduce the CPU-to-GPU memory communications and keep all the problem data on the GPU memory. Besides, the full GPU-to-GPU bandwidth can be obtained only if a coalesced memory access scheme is used (see the Nvidia documentation [97]); so, all our GPU computation kernels are implemented using that memory pattern. For our implementation of EM and SGP, two CUDA kernel libraries are very important: CUFFT and CUBLAS. The CUBLAS library is a GPU implementation of the well known BLAS library, where princi-

GPU	GT200	GF100(Fermi)	GK104(Kepler)
Transistor	1.4 Billion	3.0 Billion	3.5 Billion
Cuda core / SM	8	32	192
Cuda core (total)	240	512	1536
Graphics Core Clock	648 MHz	700 MHz	1006 MHz
Shader Core Clock	1476 MHz	1401 MHz	1006 MHz
Memory	1024 MB	1536 MB	4096 MB
Memory Clock	2484 MHz	3696 MHz	6008 MHz
Memory Bandwidth	159.0 GB/s	177.4 GB/s	192.256 GB/s
PCIe x16	2.0	2.0	3.0
Compute Capability	1.3	2.0	3.0
GFLOPs	1062.72	1344.96	3090
TDP	204 Watt	250 Watt	195 Watt

Table 3.1: A comparison between different generations of NVidia GPUs.

pal subroutines for levels 1, 2 and 3 are implemented. By the CUFFT library we can compute 1D, 2D and 3D FFTs: complex-to-complex, real-to-complex and complex-to-real versions are available. The use of these libraries is highly recommended for maximally exploiting the GPU performances.

Figure 3.1: Theoretical scheme of a Streaming Multiprocessor (SM) on the left and of a Streaming Multiprocessor eXtreme (SMX) on the right.

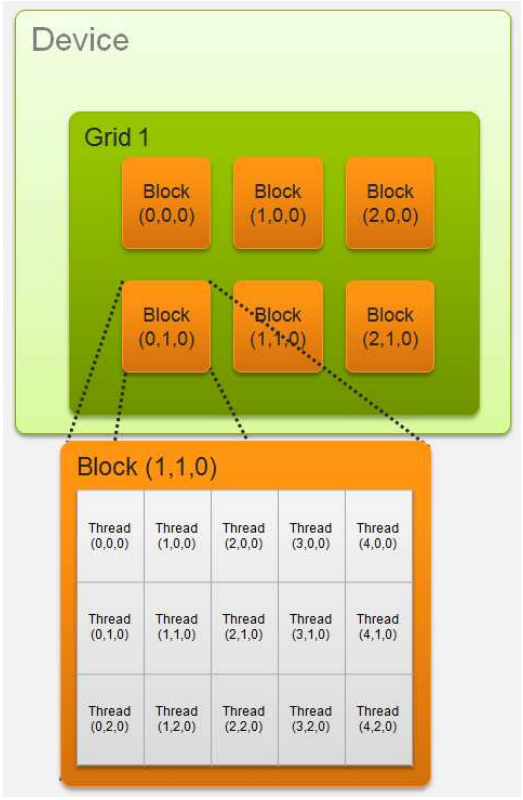


Figure 3.2: Theoretical scheme of a grid with blocks and threads

3.2.3 MPI

The second implementation is based on MPI, a language-independent communication protocol well suited for high-performance computing due to its scalability and portability.

The advantages of MPI over other message passing libraries are portability and speed. The MPI interface is meant to provide essential virtual topology, synchronization, and communication functionality between a set of processes, that usually have been mapped to nodes or computer instances. Typically, for maximum performance, each CPU (or core in a multi-core machine) will be assigned just a single process. This assignment happens at runtime through the agent that starts the MPI program, normally called mpirun or mpiexec.

MPI library functions include point-to-point send/receive operations (`MPI_Send`, `MPI_Receive`), choosing the logical process topology, exchanging data between process groups (`MPI_Bcast/MPI_Scatter`), combining partial results of computations (`MPI_Gather` and `MPI_Reduce`), synchronizing nodes (`MPI_Barrier`) as well as obtaining network-related information such as the number of processes in the computing session (`MPI_Comm_size`), current processor identity that a process is mapped to (`MPI_Comm_rank`), and so on. Point-to-point operations come in both synchronous and asynchronous forms, to allow both stronger and weaker semantics for the synchronization of processes.

In this implementation the FFT is obtained exploiting another well known library, FFTW, which is a C library for computing the discrete Fourier transform (DFT) of multidimensional real or complex data of arbitrary size. In the MPI paradigm, performance is increased by splitting the problem domain among the computational elements. Hence, the data used by the MPI FFTW routines are distributed accordingly: a distinct portion of them is locally available to each process involved in the transform. This allows the FFT to be parallelized, for instance, across a workstations cluster, each one being equipped with its own separate memory, so that one can take advantage of the total memory of all the involved processors.

As already observed, considering the deblurring problem, the main computational cost in both the EM and SGP iterations consists in a pair of forward and backward FFTs for computing the image convolutions. We face these operations by means of the CUFFT and FFTW subroutines: after computing a 2D real-to-

complex transform, the spectral multiplication between the transformed iterate and the PSF is carried out and then the 2D inverse complex-to-real transform is computed. Furthermore, both the algorithms need a componentwise division for each pixel in the image, while the computation of the objective function in SGP requires also a logarithm for each pixel. These tasks are particularly suited for both the GPU and the MPI implementation: in fact, they do not involve any dependency among the pixels and the computations can be easily distributed on all the available processors.

Concerning the denoising problem, the computation of the objective function and of its gradient are dominated by simple pixel-by-pixel operations (no image convolutions are required) that are well suited for an effective implementation on the graphics hardware [104].

For both the imaging problems, componentwise divisions and logarithms require a number of clock cycles larger than a simple floating point operation, thus the GPU memory bandwidth is not a limitation for these operations. Finally, we must discuss a critical part of the SGP algorithm: the scalar products for updating the steplength α_k from Section 2.4. The needed "reduction" operations for a scalar product imply a large number of communications among the processors and there are dependencies that prevent a straight parallelization. In our experiments, the `reduce` function provided by the Thrust library generally achieves remarkable performances while retaining sufficient stability, even in single precision: we then exploit the kernel libraries provided by Thrust, which was a separated library before the 4.0 and further CUDA release but has been included in the framework successively. In the MPI implementation, the scalar product is obtained by using the `MPI_reduction` subroutine.

Chapter 4

Gradient methods for regularization parameter estimation

As we stated before, estimating a regularization parameter can be a very difficult task, in which the solution of expensive minimization problems is required. Here we show a problem of regularization hyperparameter estimation which benefited from the same strategies introduced in Chapter 2 and that could also exploit the parallelization provided by GPU.

We are interested here in solving an inverse problem, the direct formulation of which can be written as

$$g = Au + \eta, \tag{4.1}$$

where g (vector of dimension k) denotes the observed image, A (matrix of dimension $k \times k$) denotes a linear operator (e.g. blur) which is assumed to be known and non necessarily invertible. Finally, $\eta \sim \mathcal{N}(0, \sigma^2 \mathrm{I})$ models an additive white Gaussian noise of known variance σ^2, which is usually considered in high level intensity acquisition.

Restoring u from g is an ill-posed inverse problem which must be regularized. We consider in this case regularization in the wavelet domain [41, 33, 51, 9, 120, 31],

by minimizing a convex criterion given by

$$J(u) = \frac{\|g - Au\|^2}{2\sigma^2} + \sum_{\mathbf{m}=1}^{M} \lambda_{\mathbf{m}} \phi_{\mathbf{m}}(F_{\mathbf{m}} u). \tag{4.2}$$

The first term (data fidelity) corresponds to the negative log-likelihood in the Gaussian white noise case, which brings to the mean square error, the second one is the regularization term in the wavelet domain: $F_{\mathbf{m}}u$ denotes a subband and F represents the global orthogonal wavelet transform operator [92]. Functions $\phi_{\mathbf{m}}$ model distributions of the wavelet coefficients and are chosen to be ℓ^1-norms on all the subbands. The resolution level and the number of channels (dyadic, Q-band) of the decomposition determine the finite number M of subband $\mathbf{m} = 1, ..., M$.

In such variational approach, the knowledge of $\boldsymbol{\lambda} = (\lambda_{\mathbf{m}=1,...,M})_{\mathbf{m}}$ which are called hyperparameters, is of crucial importance for the quality of the result and the question is to know how to fix these parameters from observations. In some applications and when there are only few hyperparameters, they can be empirically fixed by the user by trial and error. However, when the convergence of the $J(u)$ minimization algorithm is slow or when the number of hyperparameters to compute is more than one or two, this solution becomes difficult to apply.

Our objective is to propose a method to automatically estimate hyperparameters. Estimating $\boldsymbol{\lambda}$ only knowing a degraded observation g is the well known difficult problem of hyperparameter estimation in incomplete data [123]. Widely known state of the art methods as discrepancy principle based methods [56, 17], cross-validation methods [80] or Stein principle based approaches [120, 42] are mainly restricted to the estimation of one hyperparameter only.

Other stochastic methods such as Maximum Likelihood (ML) approaches [123, 79], EM algorithms (Expectation-Maximization) [43, 51], or MCMC (Monte Carlo Markov Chain) based sampling methods [77, 88], allow the estimation of several hyperparameters but have to face with sampling difficulties.

We adopt a ML strategy which allows us to take advantage of the good asymptotic properties of the ML estimator and to estimate a vector of hyperparameters. However, applying a gradient ascent algorithm to compute ML hyperparameter estimates is dramatically prohibitive in terms of time computing for two reasons. First, computing the gradient of the likelihood function requires to sample the a posteriori distribution, whose energy is defined by (4.2) [57]. Direct application

of Gibbs sampling and Metropolis-Hastings is not possible due to the simultaneous presence of operators A and $F_{\mathbf{m}}$ in (4.2). Second, gradient ascent methods converge slowly and much attention have to be paid to steplength determination.

Our contribution in this work can be divided in two parts. First, inspired by the approach described in [9], we propose to introduce by inference an auxiliary variable which separates operators A and F in the criterion. Second, we exploit the adaptive steplength selection previously discussed for SGP and a line-search strategy by defining a two phase algorithm increasing the convergence speed of the gradient ascent algorithm.

4.1 Maximum Likelihood estimation of the hyperparameters

4.1.1 Classical approach

The ML estimation of the vector of hyperparameters $\boldsymbol{\lambda} = (\lambda_{\mathbf{m}})_{\mathbf{m}=1,\dots,M}$ consists in maximizing $p_{\boldsymbol{\lambda}}(g)$ w.r.t $\boldsymbol{\lambda}$ where $p_{\boldsymbol{\lambda}}(g)$ is given by:

$$p_{\boldsymbol{\lambda}}(g) = \int_u p_{\boldsymbol{\lambda}}(g,u)du = \int_u p(g|u)p_{\boldsymbol{\lambda}}(u)du. \tag{4.3}$$

The integration domain of u is $[0,255]^k$ where k represents the number of elements in u (e.g. number of pixels for a 2D image).

In our context, we can easily derive that

$$p(g|u) = \frac{1}{K_\sigma}\exp\Big(\frac{-\|g-Au\|^2}{2\sigma^2}\Big) \tag{4.4}$$

where the normalization constant is $K_\sigma = (2\pi)^{k/2}\sigma^k$ and

$$p_{\boldsymbol{\lambda}}(u) = \frac{1}{Z_{\boldsymbol{\lambda}}}\prod_{\mathbf{m}=1}^{M}\exp\left(-\lambda_{\mathbf{m}}\phi_{\mathbf{m}}(F_{\mathbf{m}}u)\right) \tag{4.5}$$

where the constant $Z_{\boldsymbol{\lambda}}$ is a normalization constant defined by

$$Z_{\boldsymbol{\lambda}} = \int_u \exp\left(-\sum_{\mathbf{m}=1}^{M}\lambda_{\mathbf{m}}\phi_{\mathbf{m}}(F_{\mathbf{m}}u)\right)du. \tag{4.6}$$

The ML estimation of the vector $\boldsymbol{\lambda}$ in equation (4.3) therefore becomes by inference:

$$p_{\boldsymbol{\lambda}}(g) = \int_u p_{\boldsymbol{\lambda}}(g,u)du = \int_u p(g|u)p_{\boldsymbol{\lambda}}(u)du = \frac{Z_{\sigma,\boldsymbol{\lambda}}(g)}{K_\sigma Z_{\boldsymbol{\lambda}}} \tag{4.7}$$

where $Z_{\sigma,\boldsymbol{\lambda}}(g) = \int_u \exp\left(\frac{-\|g-Au\|^2}{2\sigma^2} - \sum_{\mathbf{m}=1}^{M} \lambda_{\mathbf{m}}\phi_{\mathbf{m}}(F_{\mathbf{m}}u)\right)du$.

The maximization of $p_{\boldsymbol{\lambda}}(g)$ with respect to $\boldsymbol{\lambda}$ can be performed using a gradient method [123] and thus requires to compute the derivatives of $p_{\boldsymbol{\lambda}}(g)$ w.r.t. each element of $\boldsymbol{\lambda}$. Invoking the Lebesgue dominated convergence theorem after the derivative we obtain:

$$\frac{\partial \log p_{\boldsymbol{\lambda}}(g)}{\partial \lambda_{\mathbf{m}}} = \frac{\partial \log Z_{\sigma,\boldsymbol{\lambda}}(g)}{\partial \lambda_{\mathbf{m}}} - \frac{\partial \log Z_{\boldsymbol{\lambda}}}{\partial \lambda_{\mathbf{m}}}$$

Where the expansion of the two terms is:

$$\begin{aligned}
\frac{\partial \log Z_{\sigma,\boldsymbol{\lambda}}(g)}{\partial \lambda_{\mathbf{m}}} &= \frac{-1}{Z_{\sigma,\boldsymbol{\lambda}}(g)} \int_u \phi_{\mathbf{m}}(F_{\mathbf{m}}u) \exp\left(\frac{-\|g-Au\|^2}{2\sigma^2} - \sum_{\mathbf{m}=1}^{M} \lambda_{\mathbf{m}}\phi_{\mathbf{m}}(F_{\mathbf{m}}u)\right)du \\
&= -\mathrm{E}_{\sigma,\boldsymbol{\lambda}}[\phi_{\mathbf{m}}(F_{\mathbf{m}}u)] \qquad (4.8) \\
\frac{\partial \log Z_{\boldsymbol{\lambda}}}{\partial \lambda_{\mathbf{m}}} &= \frac{-1}{Z_{\boldsymbol{\lambda}}} \int_u \phi_{\mathbf{m}}(F_{\mathbf{m}}u) \exp\left(-\sum_{\mathbf{m}=1}^{M} \lambda_{\mathbf{m}}\phi_{\mathbf{m}}(F_{\mathbf{m}}u)\right)du \\
&= -\mathrm{E}_{\boldsymbol{\lambda}}[\phi_{\mathbf{m}}(F_{\mathbf{m}}u)] \qquad (4.9)
\end{aligned}$$

Bringing to:

$$\frac{\partial \log p_{\boldsymbol{\lambda}}(g)}{\partial \lambda_{\mathbf{m}}} = \mathrm{E}_{\boldsymbol{\lambda}}[\phi_{\mathbf{m}}(F_{\mathbf{m}}u)] - \mathrm{E}_{\sigma,\boldsymbol{\lambda}}[\phi_{\mathbf{m}}(F_{\mathbf{m}}u)] \tag{4.10}$$

where the first expectation is defined according to the *a priori* law (4.5) and the second one, according to the *a posteriori* law

$$p_{\sigma,\boldsymbol{\lambda}}(u|g) = \frac{\exp\left(\frac{-\|g-Au\|^2}{2\sigma^2} - \sum_{\mathbf{m}=1}^{M} \lambda_{\mathbf{m}}\phi_{\mathbf{m}}(F_{\mathbf{m}}u)\right)}{\int_u \exp\left(\frac{-\|g-Au\|^2}{2\sigma^2} - \sum_{\mathbf{m}=1}^{M} \lambda_{\mathbf{m}}\phi_{\mathbf{m}}(F_{\mathbf{m}}u)\right)du}. \tag{4.11}$$

The problem here is that, contrary to the first expectation, the second expectation cannot be computed analytically and we have to generate samples according to (4.11) in order to estimate it by an empirical mean. To this end, we have to tackle the pixel dependence induced by A and this can be done by performing the calculations in the Fourier domain, where A can be diagonalized. However, this is

not sufficient as F cannot be diagonalized in the same space. Operators A and F must be split.

4.1.2 Decoupling of the linear operators

This splitting can be done by adopting the approach described in [9]. Indeed, the authors proposed to introduce an auxiliary variable w (hidden variable) and thus, they show that

$$\frac{\|g - Au\|^2}{2\sigma^2} = \min_w \frac{1}{2\sigma^2\mu}\Big(\|u - w\|^2 + \langle Cw, w\rangle\Big) + \frac{1}{2\sigma^2}\Big(\|g\|^2 - 2\langle Au, g\rangle\Big), \quad (4.12)$$

where $C = B(\mathrm{I} - B)^{-1}$ and $B = \mu A^*A$ (μ such that $\mu\|A^*A\| < 1$). This means that a new criterion can be considered instead of (4.2) which is given by

$$J(u, w) = \frac{1}{2\sigma^2\mu}\Big(\|u-w\|^2+\langle Cw, w\rangle\Big)+\frac{1}{2\sigma^2}\Big(\|g\|^2-2\langle Au, g\rangle\Big)+\sum_{\mathbf{m}=1}^{M}\lambda_{\mathbf{m}}\phi_{\mathbf{m}}(F_{\mathbf{m}}u). \quad (4.13)$$

Moreover, one can show that (4.13) can be re-expressed as:

$$\begin{aligned} J(u, w) &= \frac{1}{2\sigma^2\mu}\Big((w - (\mathrm{I} + C)^{-1}u)^{\mathrm{T}}(\mathrm{I} + C)(w - (\mathrm{I} + C)^{-1}u)\Big) + \\ &\quad + \frac{1}{2\sigma^2}\Big(\|g - Au\|^2\Big) + \sum_{\mathbf{m}=1}^{M}\lambda_{\mathbf{m}}\phi_{\mathbf{m}}(F_{\mathbf{m}}u) \end{aligned} \quad (4.14)$$

as $(\mathrm{I} + C)^{-1} = \mathrm{I} - \mu A^*A$. Note that it is shown in [9] that minimizing $J(u)$ w.r.t u is equivalent to minimize $J(u, w)$ w.r.t (u, w).

It has been shown [30] that $p_{\boldsymbol{\lambda}}(g) = \int_{u,w} p_\lambda(g, u, w)dwdu$ and we can see from (4.14) that variables g and w are independent conditionally to u. Consequently, similarly to Section (4.1.1), we can derive an ML estimation of the parameters by maximizing $p_{\boldsymbol{\lambda}}(g)$ which is now given by

$$p_{\boldsymbol{\lambda}}(g) = \int_{u,w} p(g|u)p(w|u)p_{\boldsymbol{\lambda}}(u)dudw \quad (4.15)$$

where $p(g|u)$ is still defined by (4.4), $p_{\boldsymbol{\lambda}}(u)$ is still defined by (4.5) and

$$p(w|u) = \frac{1}{K_\mu}\exp\Big(-\frac{(w - (\mathrm{I} + C)^{-1}u)^{\mathrm{T}}(\mathrm{I} + C)(w - (\mathrm{I} + C)^{-1}u)}{2\sigma^2\mu}\Big) \quad (4.16)$$

is a Gaussian law $\mathcal{N}\big((\mathrm{I}+C)^{-1}u, \sigma^2\mu(\mathrm{I}+C)^{-1}\big)$, with $K_\mu = (2\pi\sigma^2\mu)^{k/2}(\det(\mathrm{I}+C))^{-1/2}$.

The maximization of $p_{\boldsymbol{\lambda}}(g)$ with respect to each parameter $\lambda_{\mathbf{m}}$ requires also the computation of the derivatives of $p_{\boldsymbol{\lambda}}(g)$ w.r.t. each $\lambda_{\mathbf{m}}$ and leads to (invoking again the Lebesgue dominated convergence theorem):

$$\frac{\partial \log p_{\boldsymbol{\lambda}}(g)}{\partial \lambda_{\mathbf{m}}} = \mathrm{E}_{\boldsymbol{\lambda}}[\phi_{\mathbf{m}}(F_{\mathbf{m}}u)] - \mathrm{E}_{\sigma,\boldsymbol{\lambda},\mu}[\phi_{\mathbf{m}}(F_{\mathbf{m}}u)]. \tag{4.17}$$

Again, the first expectation can be computed analytically while for the second one, we have to generate samples according to the *a posteriori* law $p_{\sigma,\boldsymbol{\lambda},\mu}(u,w|g)$. Contrary to the previous case, the sampling is now possible in reasonable computing time as $J(u,w)$ can be rewritten as (4.13). In this second configuration, variables u and w are now decoupled and they can be both estimated in two decorrelated spaces (wavelets for u, Fourier for w). A is applied to g and w. Only F remains applied to u which is not a problem to sample according to $p_{\sigma,\boldsymbol{\lambda},\mu}(u,w|g)$.

The sampling according to the a posteriori law can be done using a two steps algorithm that alternates a Gibbs sampler and a Metropolis Hastings procedure [102]. Indeed, to sample according to $p_{\sigma,\boldsymbol{\lambda},\mu}(u,w|g)$:

1. We first generate (Gibbs sampler) samples according to $p(w|u)$ given by (4.16) (Gaussian law). The variable w is directly expressed in the Fourier transform domain as the covariance matrix can be diagonalized easily;

2. Secondly, having a generation of w samples, we generate u samples or more precisely, directly wavelet transform coefficients (Metropolis Hastings algorithm) according to $p_{\boldsymbol{\lambda}}(u|w,g)$ where:

$$p_{\boldsymbol{\lambda}}(u|w,g) \propto \exp\Big(-\frac{1}{2\sigma^2\mu}\|Fu-Fw\|^2 + \frac{1}{\sigma^2}\langle Fu, FA^*g\rangle - \sum_{\mathbf{m}}\lambda_{\mathbf{m}}\phi_{\mathbf{m}}(F_{\mathbf{m}}u)\Big). \tag{4.18}$$

4.2 Proposed algorithms

4.2.1 Classical gradient ascent

As mentioned previously, parameters $\lambda_{\mathbf{m}}$ are computed by launching a gradient ascent (GA) algorithm [123] which can be written here as, $\forall \mathbf{m} \in \{1, \ldots, M\}$,

$$\lambda_{\mathbf{m}}^{(n+1)} = \lambda_{\mathbf{m}}^{(n)} + \alpha_n \left[\mathrm{E}_{\boldsymbol{\lambda}}[\phi_{\mathbf{m}}(F_{\mathbf{m}} u)] - \frac{2}{L} \sum_{l=L/2+1}^{L} \phi_{\mathbf{m}}\big(F_{\mathbf{m}} {u^{(n)}}_{\sigma,\boldsymbol{\lambda}^{(n)},\mu}\big)_l) \right]. \quad (4.19)$$

The first expectation is computed analytically (closed-form expression for the chosen $\phi_{\mathbf{m}}$) and $\big({u^{(n)}}_{\sigma,\boldsymbol{\lambda}^{(n)},\mu}\big)_l$ denotes the l-th sample generated according to the *a posteriori* probability density $p_{\boldsymbol{\lambda}^{(n)}}(u, w|g)$. Here, L denotes the number of computed samples and $L/2$ samples are required to initialize the chain. The parameter α_n represents here the steplength of the algorithm and can vary along with the iterations. The choice of this steplength is crucial and directly governs the algorithm convergence. For this reason, we decided to pay much attention to it as described later.

ALGORITHM 4.1 Gradient Ascent

Set $\mu < 1/\|A^*A\|$
Set $u^{(-1)} = g$
for each subband **m** **do**
  Initialize $\lambda_{\mathbf{m}}^{(0)}$.
end for
for $n = 0, 1, \ldots$ **do**
  Fix an algorithm steplength α_n
  for $l = 0, 1, \ldots, L$ **do**
    Generate w_l directly in the frequency domain according to $p(w|u^{(n-1)})$ as described in Algo. 4.2
    for each subband **m** **do**
      Generate $\left(F_{\mathbf{m}} u^{(n)}_{\sigma,\boldsymbol{\lambda}^{(n)},\mu}\right)_l$ in the wavelet transform domain according to $p(u^{(n-1)|w_l,g})$ as described in Algo. 4.3
    end for
  end for
  for each subband **m** **do**
    $\lambda_{\mathbf{m}}^{(n+1)} = \lambda_{\mathbf{m}}^{(n)} + \alpha_n \left[\mathrm{E}_{\boldsymbol{\lambda}}[\phi_{\mathbf{m}}(F_{\mathbf{m}} u)] - \frac{2}{L} \sum_{l=L/2+1}^{L} \phi_{\mathbf{m}}\big(F_{\mathbf{m}} {u^{(n)}}_{\sigma,\boldsymbol{\lambda}^{(n)},\mu}\big)_l) \right]$.
  end for
end for

ALGORITHM 4.2 Gibbs Sampler: generation of w samples according to $p(w|u)$

- Given u, compute
  mean $(\mathrm{I}+C)^{-1}u = (\mathrm{I}-\mu A^*A)u$
  variance $\sigma^2\mu(\mathrm{I}-\mu A^*A)$
  of the Gaussian distribution
- Generate first w as $\mathcal{N}(0,1)$ and then compute the Fourier transform of w.
- Restore the mean of the distribution and then the variance.
- w is obtained applying an inverse Fourier transform of the obtained coefficients.

ALGORITHM 4.3 Metropolis-Hastings: generation of u samples according to $p_{\boldsymbol{\lambda}}(u|w,g)$

- Let $k_{\mathbf{m}}$ be the subband size
- Fix the proposal density to $\mathcal{N}(0,\sigma_p^2)$ and generate coefficients $F_{\mathbf{m}}c$ according to this law.
- Generate $k_{\mathbf{m}}$ uniformly distributed random values v in $[0,1]$.

  if $\frac{p_{\boldsymbol{\lambda}}(c|w,g)}{p_{\boldsymbol{\lambda}}(u|w,g)} > v$ (component-wise computation) **then**
  $u = c$ (component-wise computation)
  end if

4.2.2 Acceleration techniques and line-search strategies

In order to accelerate gradient methods, an effective technique consists in using adaptive steplength rules for defining the step α_k along the gradient direction, combined with line-search strategies that, if necessary, shorten the step for ensuring suited improvements in the objective function. We exploit these ideas also for designing an accelerated gradient approach for maximizing $L(\boldsymbol{\lambda}) = \log p_{\boldsymbol{\lambda}}(g)$. The main difficulty is due to the fact that the objective function defined in (4.15) can't be evaluated and, as a consequence, standard line-search strategies are not useful. To overcome this difficulty we develop a two phases gradient method in

which, firstly, a sequence of simple gradient steps are performed with the aim to improve the objective function and, secondly, a line-search that avoids the use of the objective function is introduced in the iterative process, for reducing the gradient norm of the objective function.

The first phase consists in steps of the form (4.19) in which the steplength is obtained by an adaptive alternation of the well known Barzilai-Borwein values [8]:

$$\alpha_k^{BB1} = -\frac{s_k^T s_k}{s_k^T y_k}, \qquad \alpha_k^{BB2} = -\frac{s_k^T y_k}{y_k^T y_k},$$

where $s_k = \boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^{(k-1)}$ and $y_k = \nabla L(\lambda^{(k)}) - \nabla L(\lambda^{(k-1)})$. The adaptive alternation is derived from [128, 53] and described in [21].

In the second phase a line-search strategy similar to the one proposed in [32] is used. In the following we introduce the main properties on which the line-search strategy is based.
Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a continuous differentiable mapping (in the case of the hyperparameters estimation, $F(x)$ can be viewed as the gradient of $L(\boldsymbol{\lambda})$), assume the Jacobian J of F symmetric for every $x \in \mathbb{R}^n$ and let f be the norm function $f(x) = \frac{1}{2}\|F(x)\|^2$.

In [32, 62] iterative methods for the minimization of $f(x)$ are stated; they generate a sequence of iterates $\{x_k\}$ in which $x_{k+1} = x_k + \xi_k d_k$, where d_k denotes a movement direction and $\xi_k > 0$ is the corresponding line-search parameter. The direction is defined as $d_k = -q_k(\alpha_k)$ where:

$$q_k(\alpha_k) = \frac{F(x_k + \alpha_k F(x_k)) - F(x_k)}{\alpha_k}, \qquad \alpha_k > 0.$$

First of all we observe that, under the above assumptions,

$$\lim_{\alpha_k \to 0} q_k(\alpha_k) = J(x_k)^T F(x_k) = \nabla f(x_k). \tag{4.20}$$

Therefore, when α_k is sufficiently small, the vector $q_k(\alpha_k)$ is a good approximation of $\nabla f(x_k)$, consequently a line-search that is norm descent can be exploited to determine a steplength$\xi_k > 0$ which satisfies

$$\|F(x_k + \xi_k d_k)\| \leq \|F(x_k\|.$$

We formalize this idea in the following lemma.

Lemma 4.1 *If $\nabla f(x_k) \neq 0$, then there exists a constant $\bar{\alpha}_k$ such that when $\alpha_k \in (0, \bar{\alpha}_k)$,*

$$-\nabla f(x_k)^T q_k(\alpha_k) < 0. \tag{4.21}$$

Moreover, given a positive constant γ, the inequality

$$f(x_k - \alpha_k^2 q_k(\alpha_k)) - f(x_k) \leq -\gamma \|\alpha_k^2 F(x_k)\|^2 \tag{4.22}$$

holds for all $\alpha_k > 0$ sufficiently small.

Proof. By (4.20) we have that

$$\begin{aligned} -\lim_{\alpha_k \to 0^+} \nabla f(x_k)^T q_k(\alpha_k) &= -F(x_k)^T J(x_k) J(x_k)^T F(x_k) \\ &= -\|J(x_k)^T F(x_k)\|^2 \end{aligned}$$

and, since $J(x_k)^T F(x_k) \neq 0$, we obtain (4.21). Notice that

$$\lim_{\alpha_k \to 0^+} \frac{f(x_k - \alpha_k^2 q_k(\alpha_k)) - f(x_k)}{\alpha_k^2} = \lim_{\alpha_k \to 0^+} -\nabla f(x_k)^T q_k(\alpha_k) = -\|J(x_k)^T F(x_k)\|^2 < 0$$

The right-hand side of (4.22) is $o(\alpha_k^2)$, therefore inequality (4.22) holds for all $\alpha_k > 0$ sufficiently small. □

The above Lemma can be exploited for designing a special iterative scheme for minimizing $f(x)$ that ensures, for sufficiently small α_k, nonmonotone descent steps for $f(x)$ by first trying a movement along the direction $F(x_k)$ and, if this is not successful, by moving along the direction $-q_k(\alpha_k)$. Thus, when $F(x_k)$ denotes the gradient of an objective function $\phi(x)$ to be maximized (as in the case of ML estimation of the hyperparameters), such an iterative approach can try steps along the gradient of $\phi(x)$ while ensuring the nonmonotone reduction of $\|F(x_k)\|$.

The proposed iterative scheme for minimizing $f(x)$ and its line-search strategy can be described as follows: given an integer $P > 0$, parameters $\theta, \gamma \in (0, 1)$ and a tentative step-length $\alpha_k, 0 < \alpha_{min} \leq \alpha_k \leq \alpha_{max}$, define $x_{(k+1)}$ by Algorithm 4.4

ALGORITHM 4.4 Updating Step

IF $f\big(x_k + \alpha_k F(x_k)\big) \leq \max\limits_{0\leq j\leq \min(k,P-1)} f(x_{(k-j)}) - \gamma\|\alpha_k^2 F(x_k)\|^2$ THEN

set $x_{(k+1)} = x_k + \alpha_k F(x_k)$ and $k = k + 1$;

ELSE IF $f\big(x_k - \alpha_k^2 q_k(\alpha_k)\big) \leq \max\limits_{0\leq j\leq \min(k,P-1)} f(x_{(k-j)}) - \gamma\|\alpha_k^2 F(x_k)\|^2$ THEN

set $x_{(k+1)} = x_k - \alpha_k^2 q_k(\alpha_k)$ and $k = k + 1$;

ELSE

set $\alpha_k = \theta\alpha_k$ and go back to the IF statement;

ENDIF

It is interesting to remark that:

1. the trial point $x_k + \alpha_k F(x_k)$ could not satisfy the nonmonotone line-search condition even for small α_k; however, from Lemma 4.1, after a finite number of reduction of the steplength α_k, the second condition necessarily holds. This means that the line-search process is well defined.

2. the line-search process ensures that the nonmonotone norm-descent property is satisfied even if the directions $F(x_k)$ or $-q_k(\alpha_k)$ are not guaranteed to be descent directions of f at x_k. In the case $P = 1$, the line-search strategy forces a monotone descent of the sequence $\{f(x_k)\}$.

This line-search strategy allows to prove that every limit point x^* of $\{x_k\}$ is a stationary points of $f(x)$, that is $\|\nabla f(x^*)\| = \|J(x^*)^T F(x^*)\| = 0$; in particular, if $J(x^*)$ is nonsingular, such a stationary point satisfies $\|F(x^*)\| = 0$ and, if $F(x) = \nabla L(x)$, we have that x^* is a stationary point of $L(x)$.
To obtain this convergence property we need some other assumptions:

Assumption 1.

(a) The level set $\Omega = \{x \in \mathbb{R}^n : 0 \leq f(x) \leq f(x_0)\}$ is bounded.

(b) In some neighbourhood Γ of Ω, the nonlinear mapping $F(x)$ has continuous partial derivatives and is Lipschitz continuous, therefore there exists a

constant $L > 0$ such that

$$\|F(x) - F(y)\| \leq L\|x - y\|, \forall x, y \in \Gamma \tag{4.23}$$

From assumption (a) we can assert that there exists a positive constant δ such that

$$\|F(x)\| \leq \delta, \forall x \in \Omega. \tag{4.24}$$

Under the above assumptions, it is possible to prove the following convergence property by proceeding as in [32]. We firstly state some preliminary definitions. Define $V_0 = f(x_0)$ and

$$V_k = \max f(x_{(k-1)P+1}), \ldots, f(x_{kP}), \quad \forall k = 1, 2, \ldots$$

Let $\nu_{(k)} \in \{(k-1)P + 1, \ldots, kP\}$ be such that for all $k = 1, 2, \ldots,$

$$f(x_{\nu(k)}) = V_k.$$

We recall from [82] the following two lemmas.

Lemma 4.2 *For all $k, l \in N$, we have*

$$f(x_{kP+l}) \leq f(x_{\nu(k)})$$

Lemma 4.3 *Suppose the Assumption 1 holds. Let $\{x_k\}$ be generated by Algorithm 4.4. Then we have*

$$\lim_{k\to\infty} \alpha^4_{\nu(k)-1} f(x_{\nu(k)-1}) = 0.$$

From now on, we define $K = \{\nu(1) - 1, \nu(2) - 1, \nu(3) - 1, \ldots\}$. The following theorem establishes the global convergence of the iterative method based on the updating step described in Algorithm 4.4.

Theorem 4.1 *Let $\{x_k\}$ be generated with the Algorithm 4.4; then every limit point x^* of $\{x_k\}_K$ satisfies*

$$J(x^*)^T F(x^*) = 0.$$

Proof. From Lemma 4.3 we can obtain

$$\lim_{k\in K} \alpha_k^4 \|F(x_k)\|^2 = 0 \tag{4.25}$$

Let x^* be a limit point of $\{x_k\}_K$. Let $K_1 \subset K$ be a set such that

$$\lim_{k\in K_1} x_k = x^*.$$

Consequently we have

$$\lim_{k\in K_1} \alpha_k^4 \|F(x_k)\|^2 = 0.$$

Consider the two cases $\lim_{k\in K_1} \alpha_k \neq 0$ and $\lim_{k\in K_1} \alpha_k = 0$.

Case 1: If $\lim_{k\in K_1} \alpha_k \neq 0$, there exists a set $K_2 \subset K_1$ such that α_k is bounded away from zero for $k \in K_2$. Then by (4.25),

$$\lim_{k\in K_2} \|F(x_k)\|^2 = 0.$$

Since F is continuous and $\lim_{k\in K_2} x_k = x^*$ this implies that $F(x^*) = 0$.

Case 2: If $\lim_{k\in K_1} \alpha_k = 0$ by the updating rule described in Algorithm 4.4 we have

$$f\left(x_k - \frac{\alpha_k^2}{\rho^2} q_k\left(\frac{\alpha_k}{\rho}\right)\right) - f(x_k) \geq f\left(x_k - \frac{\alpha_k^2}{\rho^2} q_k\left(\frac{\alpha_k}{\rho}\right)\right) - \max_{0\leq j\leq \min\{k,P-1\}} f(x_k)$$

$$> -\gamma \frac{\alpha_k^4}{\rho^4} \|F(x_k)\|^2$$

From (4.24) we obtain

$$\frac{f\left(x_k - \frac{\alpha_k^2}{\rho^2} q_k\left(\frac{\alpha_k}{\rho}\right)\right) - f(x_k)}{\frac{\alpha_k^2}{\rho^2}} > -\gamma\delta^2 \frac{\alpha_k^2}{\rho^2}$$

By the mean-value theorem, here also should exists a constant $\xi_k \in (0, 1)$ such

that

$$\left\langle J\left(x_k-\xi_k\frac{\alpha_k^2}{\rho^2}q_k\left(\frac{\alpha_k}{\rho}\right)\right)^T F\left(x_k-\xi_k\frac{\alpha_k^2}{\rho^2}q_k\left(\frac{\alpha_k}{\rho}\right)\right),-q_k\left(\frac{\alpha_k}{\rho}\right)\right\rangle > -\gamma\delta^2\frac{\alpha_k^2}{\rho^2} \tag{4.26}$$

By using (4.23) and (4.24) we have

$$\left\|q_k\left(\frac{\alpha_k}{\rho}\right)\right\| = \frac{\left\|F(x_k+\frac{\alpha_k}{\rho}F(x_k))-F(x_k)\right\|}{\frac{\alpha_k}{\rho}} \leq L\delta.$$

Then

$$\lim_{k\in K_1}\frac{\alpha_k^2}{\rho^2}q_k\left(\frac{\alpha_k}{\rho}\right)=0$$

and, by taking limits in (4.26) and by using (4.20), we obtain

$$J(x^*)^T F(x^*)=0.$$

□

The following theorems states that if there exists a limit point x^* of $\{x_k\}$ that is solution of $F(x)=0$, then all the limit points of $\{x_k\}$ are solution of the same problem. Moreover, if x^* is an isolated solution, then the whole sequence $\{x_k\}$ converges to x^*.

Theorem 4.2 *Let $\{x_k\}$ be generated with the Algorithm 4.4. If there exists a limit point of x^* of the sequence $\{x_k\}$ such that $F(x^*)=0$, then*

$$\lim_{k\to\infty}F(x_k)=0.$$

Theorem 4.3 *Let $\{x_k\}$ be generated with the Algorithm 4.4. Suppose that there exists a limit point x^* of the sequence $\{x_k\}$ such that $F(x^*)=0$ and there exists $\delta>0$ such that $F(x)\neq 0$ whenever $0<\|x-x^*\|\leq\delta$. Then*

$$\lim_{k\to\infty}x_k=x^*.$$

The proofs of Theorems 4.2 and 4.3 can be obtained by proceeding as in [32].

4.2.3 Two-phases (2Ph) gradient method

Denoting $G(\boldsymbol{\lambda}) = \nabla L(\boldsymbol{\lambda})$ and $f(\boldsymbol{\lambda}) = \frac{1}{2}\|G(\boldsymbol{\lambda})\|^2$, Algorithm 4.4 allows to sufficiently reduce $f(\boldsymbol{\lambda})$ by performing an ascent gradient step or moving along the direction defined by $-q_n(\alpha_n) = -\dfrac{(G(\boldsymbol{\lambda}^{(n)} + \alpha_n G(\boldsymbol{\lambda}^{(n)})) - G(\boldsymbol{\lambda}^{(n)}))}{\alpha_n}$ (for sufficiently small α_n). Due to this reduction property, in we have proven that if there exists a limit point $\boldsymbol{\lambda}^*$ of $\{\boldsymbol{\lambda}^{(n)}\}$ such that $G(\boldsymbol{\lambda}^*) = 0$, then all the limit points of $\{\boldsymbol{\lambda}^{(n)}\}$ solve $G(\boldsymbol{\lambda}) = 0$. Therefore, after a first phase in which simple BB-like gradient steps are performed, a second phase can be exploited for stabilizing our iterative process by forcing the approximation of a stationary point of $f(\boldsymbol{\lambda})$. The switching between the two phases is performed with the aim to activate the Algorithm 4.4 when the last iterations lay in a region in which $f(\boldsymbol{\lambda})$ is concave. By recalling that when $s_n^T y_n > 0$ we may conclude that the function $p_{\boldsymbol{\lambda}}(g)$ is not concave in a set containing $\boldsymbol{\lambda}^{(n)}, \boldsymbol{\lambda}^{(n-1)}$, we activate the second phase when a sequence of N_2 consecutive iterations always provide positive BB steplengths. Moreover, we ensure the switching to the second phase after a prefixed number $N_1 > N_2$ of iterations. The two phases algorithm can be described as in Algorithm 4.5.

ALGORITHM 4.5 Two Phases (2Ph) Gradient Method

Initialization: choose $\boldsymbol{\lambda}^{(0)}$, α_0 and $\theta, \gamma \in (0,1)$; set flag(N_1,N_2)=0, $n = 1$, $\boldsymbol{\lambda}^{(1)} = \boldsymbol{\lambda}^{(0)} + \alpha_0 G(\boldsymbol{\lambda}^{(0)})$, $gr_0 = f(\boldsymbol{\lambda}^{(0)})$, $gr = f(\boldsymbol{\lambda}^{(1)})$ and an integer $P \geq 1$.

Phase 1: (BB-like Gradient step)

WHILE $\left(\dfrac{gr}{gr_0} > \tau_g \text{ or } \dfrac{\|\boldsymbol{\lambda}^{(n)} - \boldsymbol{\lambda}^{(n-1)}\|}{\|\boldsymbol{\lambda}^{(n)}\|} > \tau_\lambda \right)$ and flag(N_1,N_2) $= 0$

1.1 Choose α_n and update flag(N_1,N_2) (See Alg. 4.6);

1.2 Gradient Step: $\boldsymbol{\lambda}^{(n+1)} = \boldsymbol{\lambda}^{(n)} + \alpha_n G(\boldsymbol{\lambda}^{(n)})$, $n = n + 1$;

1.3 Set $gr = \max_{0 \leq j \leq \min(n, P-1)} f(\boldsymbol{\lambda}^{(n-j)})$;

ENDWHILE

Phase 2: (Stabilization with line-search)

WHILE $\left(\dfrac{gr}{gr_0} > \tau_g \text{ or } \dfrac{\|\boldsymbol{\lambda}^{(n)} - \boldsymbol{\lambda}^{(n-1)}\|}{\|\boldsymbol{\lambda}^{(n)}\|} > \tau_\lambda \right)$

2.1 Choose α_n (See Alg. 4.6);

2.2 Line-search:

IF $f\big(\boldsymbol{\lambda}^{(n)} + \alpha_n G(\boldsymbol{\lambda}^{(n)})\big) \leq gr - \gamma \|\alpha_n^2 G(\boldsymbol{\lambda}^{(n)})\|^2$ THEN
set $\boldsymbol{\lambda}^{(n+1)} = \boldsymbol{\lambda}^{(n)} + \alpha_n G(\boldsymbol{\lambda}^{(n)})$ and $n = n + 1$;
ELSE IF $f\big(\boldsymbol{\lambda}^{(n)} - \alpha_n^2 q_n(\alpha_n)\big) \leq gr - \gamma \|\alpha_n^2 G(\boldsymbol{\lambda}^{(n)})\|^2$ THEN
set $\boldsymbol{\lambda}^{(n+1)} = \boldsymbol{\lambda}^{(n)} - \alpha_n^2 q_n(\alpha_n)$ and $n = n + 1$;
ELSE
set $\alpha_n = \theta \alpha_n$ and go to Step 2.2;
ENDIF

2.3 Set $gr = \max_{0 \leq j \leq \min(n, P-1)} f(\boldsymbol{\lambda}^{(n-j)})$;

ENDWHILE

ALGORITHM 4.6 2Ph Steplength Selection

Given the parameter $\tau \in (0,1)$ and the integers $N_1 > N_2 \geq 1$, $r \geq 1$ and $s_n = \boldsymbol{\lambda}^{(n)} - \boldsymbol{\lambda}^{(n-1)}$ and $y_n = \nabla L(\boldsymbol{\lambda}^{(n)}) - \nabla L(\boldsymbol{\lambda}^{(n-1)})$.

IF $s_n^T y_n \geq 0$ THEN

$\alpha_n^{BB1} = \min\left\{\alpha_{\max}, \max\left\{2\alpha_{n-1}^{BB1}, \alpha_{\min}\right\}\right\}$,
$\alpha_n^{BB2} = \min\left\{\alpha_{\max}, \max\left\{2\alpha_{n-1}^{BB2}, \alpha_{\min}\right\}\right\}$,
IF ($n < N_1$ and $ind < N_2$) THEN
$ind = 0$
ELSE
flag(N_1,N_2) = 1
ENDIF

ELSE

$$\alpha_n^{BB1} = \min\left\{\alpha_{\max}, \max\left\{-\frac{s_n^T s_n}{s_n^T y_n}, \alpha_{\min}\right\}\right\},$$
$$\alpha_n^{BB2} = \min\left\{\alpha_{\max}, \max\left\{-\frac{s_n^T y_n}{y_n^T y_n}, \alpha_{\min}\right\}\right\},$$

IF ($n < N_1$ and $ind < N_2$) THEN
$ind = ind + 1$
ELSE
flag(N_1,N_2) = 1
ENDIF

ENDIF

IF $\alpha_n^{BB2}/\alpha_n^{BB1} > \tau$ THEN
$\alpha_n = \alpha_n^{BB1}$
$\tau = \tau \cdot 1.1$

ELSE
$\alpha_n = \min\left\{\alpha_j^{BB2} \mid j = \max\{1, n-r+1\}, \ldots, n\right\}$
$\tau = \tau \cdot 0.9$

ENDIF

Chapter 5

Numerical Experiments

5.1 Test problems and test platforms

In this chapter we want to compare the behaviour of the different implementations of the SGP algorithm with respect to serial or parallel implementations of it and of different state of the art methods.

Due to the temporal extension of the study, we want to stress that different test suites were executed on different generations of architectures, mainly because of the extremely fast growing and developing of technology. Therefore, the comparison between image reconstruction time from different tests may not be significant. Nevertheless the speedup obtained and computed with respect to each test is consistent.

For a discussion as thoroughly as possible, for each test phase will be specified the hardware on which the experiments have been carried out and the description of the test set.

5.2 Preliminary study

Firstly, single image deblurring and denoising are performed with the Matlab,C++, CUDA and MPI versions of the code. This was the first study made to compare the different implementations and is reported in [29].

To evaluate the effectiveness of the proposed parallel implementations, we consider two sets of experiments: one for 2D images and one for a 3D object. In all these tests the SGP algorithm operates in monotone mode, that is $M = 1$.

In the 2D cases we use some deblurring problems on astronomical images and denoising problems on synthetic data. The former have pixel values in the range $\left[6.3 \cdot 10^{-9}, 1.2 \cdot 10^{-8}\right]$, being photon counters normalized by 10^{12}. The latter have integer values in the range $[0, 263]$. In both cases the images are corrupted by Poisson noise.

For the deblurring case, we look for a regularized solution of (1.2)–(1.3) by early stopping the SGP method applied to the minimization of the KL divergence (1.10). In the astronomical images we take a constant background $b = 6.76 \cdot 10^{-9}$ for each pixel.

In the 3D case we use a set of real-world data arising from a microscopy application, to give an idea of what actually happens.

Our test platform consists of a workstation equipped with 2 Intel Xeon E5620 QuadCore CPUs at 2.4GHz, 18GB of RAM and 1 GPUs Nvidia Tesla C2050 graphics card, which has 14 streaming multiprocessors (448 total cores) running at 1.15 GHz. The total amount of global memory is 3 GB and the connection bus with the cores has a bandwidth of 148 GB/sec. The peak computing performance is 1.03 Tflops/sec for single precision and 515 Gflops/sec for double precision arithmetic. We consider two CPU implementations of SGP: one in Matlab v. 7.11.0 and another one in C++. The GPU implementations are developed in mixed C and CUDA languages (CUDA 3.1 version), within Microsoft Visual Studio 2005. Finally, we run the MPI implementation on the IBM SP6 cluster at CINECA (`http://www.cineca.it/it/node/776`).

Since we consider different implementations, in different languages, on different machines, it is hard to give meaningful comparisons other than the absolute total computational time. Nevertheless, we checked the computational power of the single CPU of the IBM SP6 and that of the workstation. We ran the same C++ code, compiled with the same settings on both machines, up to 100 SGP iterations on the same test problem: we got 1.45 seconds and 1.40 seconds, respectively, for the 256×256 test, while for a test sized 1024×1024 we got 29.37 and 28.65 seconds, respectively. Hence, the computational power of the two CPUs is essentially the same and the times reported in the following tables are consistent.

The deblurring test problems are generated as follows (see [23] for more details): we convolve the original 256×256 image in Figure 5.1A with an ideal PSF, then we add a constant background term and we perturb the resulting image with Poisson

N. proc.	1	4	8	16	32	64	Matlab	GPU
SGP (sec)	4.2	1.9	1.0	0.5	0.3	0.2	15.9	0.3
RL (sec)	40.4	16.2	9.8	5.4	3.2	2.4	336.3	6.3

Table 5.1: Computing times for test problem 5.1C, sized 1024×1024.

noise to simulate real observed data (Figure 5.1C). To obtain larger test problems, we expand the original images and the PSF by means of a zero-padding technique on their FFTs. The expansion is made by preserving the medium value of the pixels and by using the same value of the background; as a consequence, the noise levels of the new larger images are comparable with those of the corresponding blurred noisy images sized 256×256. In this way, from the test problem 5.1A, we derive other test problems with sizes 512×512, 1024×1024, 2048×2048 and 4096×4096, on which the scaling properties of the iterative reconstruction algorithms can be evaluated.

For the denoising tests the original image is the LCR-phantom in Figure 5.1B, consisting of square-enclosed circles with different intensities. This image is then perturbed by Poisson noise to give the synthetic data of Figure 5.1D (see [86] and [124] for additional insights). We underline that, when we solve the deblurring problems, in all the implementations we use the number of iterations that minimizes the reconstruction error. Both RL and SGP get to the same error level, but SGP gets its minimum reconstruction error with many fewer iterations.

For denoising problems we empirically determine the optimal value of the regularization parameter μ, giving the minimal reconstruction error. We then run all the SGP implementations until the same stopping rule is satisfied. We do not report here the details of these error evaluations.

In Table 5.1 we observe the reconstruction time for the 1024×1024 deblurring test image. The computational speedup is obtained in terms of both language implementation (C++ versus Matlab) and parallelization (CUDA versus MPI). In the case of the MPI code, we can notice a saturation effect: between 32 and 64 processors, the speedup decreases because communications dominate computations.

Table 5.2 reports the times for the 4096×4096 denoising test. Looking at the MPI experiments, we can see that the saturation does not occur even up to 256 processors, while in the GPU code the increased number of pixels gives the opportunity to fully exploit all the cores and minimize the memory latency.

N. proc.	1	4	8	16	32	64	128	256	GPU
SGP (sec)	384.5	132.2	64.9	31.8	15.9	8.1	4.2	2.4	12.2

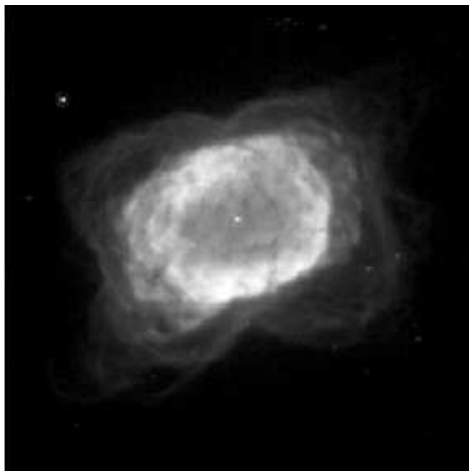
Table 5.2: Computing times for test image 5.1D, sized 4096×4096.

N. proc.	1	4	8	16	32	GPU
SGP (sec)	46.1	12.5	6.4	3.5	1.7	7.3
RL (sec)	146.2	39.7	19.0	11.7	6.0	36.5

Table 5.3: Deblurring times for 3D test. Object size: $256 \times 256 \times 52$.

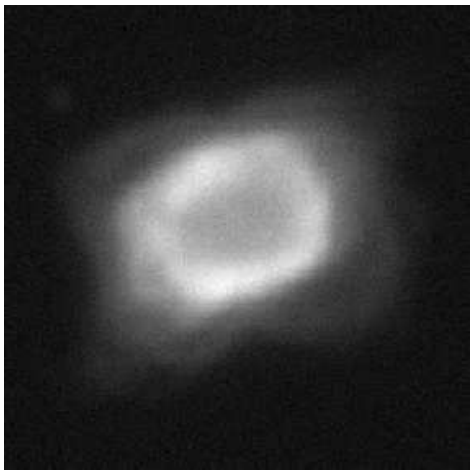
Last, we want to face a huge-size problem: a blurred multi-dimensional microscopy image sized $256 \times 256 \times 52$, showing a trait of β-tubulin protein. The data values are integers in the range $[0, 167]$ with no background. Here we do not have the original image to compare with: so, we use the classical RL algorithm to obtain a suitable degree of *visual* enhancement. This is obtained after about 300 iterations. Hence we let RL perform exactly this amount of iterations, then we run SGP until a very similar reconstruction is obtained. We found that the SGP reconstruction having the minimum relative Euclidean distance to the RL reconstruction is obtained after only 50 iterations, up to a relative tolerance of 5% on the Euclidean norm of the difference. We also checked the relative uniform norm of the difference: this is larger than the 5% threshold in only a very small part (0.007%) of all the 3.4 Mvoxels of the problem.

The visual results on some slices parallel to the cartesian planes are shown in Figures 5.3 and 5.4, where the reference system for the observed volume is supposed to be oriented as in Figure 5.2. The computational times are in Table 5.3: it can be clearly seen how in this larger case the MPI-based cluster implementation outperforms the GPU implementation very soon (between 6 and 8 processors). This is because with such large-scale data the GPU runs out of local memory and heavy data transfer are needed from/to the main memory.

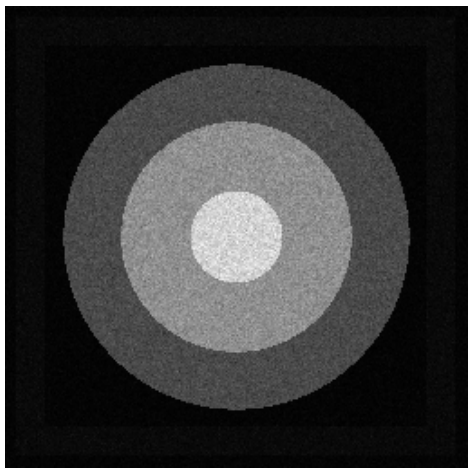


(A) Original: a galaxy.

(B) Original: the LCR phantom.



(C) Blurred noisy galaxy.



(D) Noisy LCR phantom.

Figure 5.1: 2D test data. Upper panels: original images. Lower panels: artificially perturbed images.

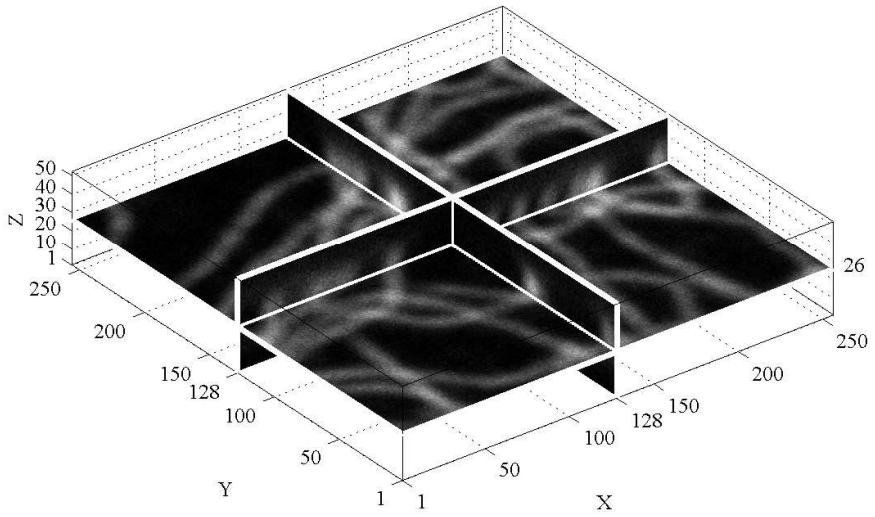


Figure 5.2: Cartesian reference system for the 3D object test problem. The volume consists of $256 \times 256 \times 52$ voxels. One can see the actual positions of the slices shown in Figures 5.3 and 5.4.

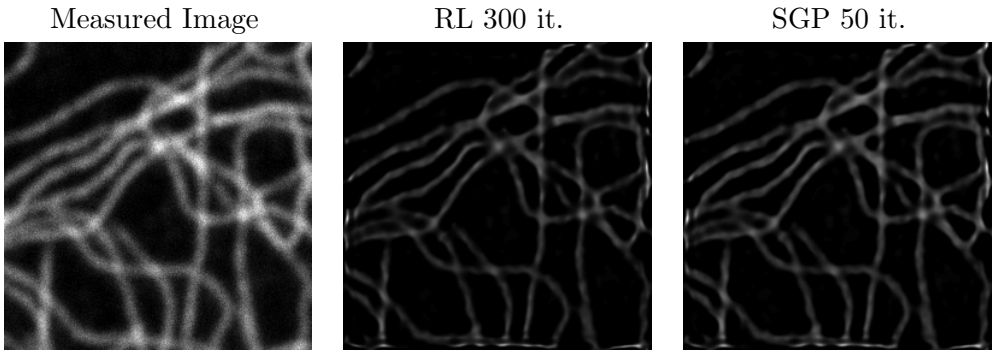


Figure 5.3: 3D object and reconstructions: slices along the transverse plane (XY), taken at the center of the volume (slices number 26 along the Z direction). Each image is sized 256×256.

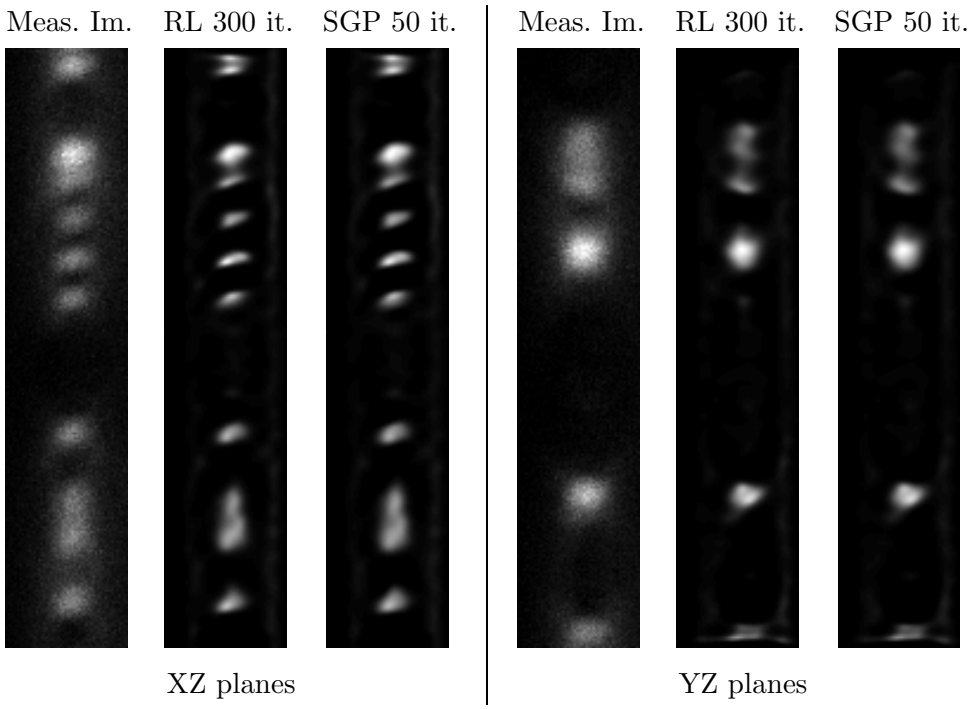


Figure 5.4: 3D object and reconstructions: slices along the frontal (XZ) and sagittal (YZ) planes, but rotated 90 degrees counterclockwise for easier picturing purpose. The slices are taken at the center of the volume (slices number 128 along the Y and the X direction, respectively). Each image is sized 256×52.

5.3 Astronomy: deconvolution with LBT

Secondly, after the promising results obtained in the previous Section, we decided to develop an IDL version of the code to be able to reach the astronomical environment, in which this language is well known and problems that needs deconvolution arise as explained in Chapter 1.

The SGP, RL and OSEM algorithms are implemented in IDL and the codes have been freely distributed. Moreover SGP implementation for GPU (graphic processor unit) is also provided. In this study we consider only the constraint of non-negativity. In Bonettini et al [23], the case of both non-negativity and flux conservation is investigated and an efficient algorithm, for the computation of the projection on the convex set defined by the constraints, is given. However their numerical experiments seem to demonstrate that the additional flux constraint does not improve significantly the reconstructions.

In the next Sections we demonstrate, by a few numerical experiments, the effectiveness of the SGP algorithm and its IDL–based GPU implementation in the solution of the deblurring problems described in Chapter 1. Our test platform consists in a personal computer equipped with an AMD Athlon X2 Dual-Core at 3.11GHz, 3GB of RAM and the graphics processing unit NVIDIA GTX 280 with CUDA 3.2. We consider CPU implementations of RL, SGP and OSEM in IDL 7.0; the GPU implementations are developed in mixed IDL and CUDA language by means of the GPUlib 1.4.4.

The set of numerical experiments is divided in two groups: single image and multiple image deconvolution. For each group, some tests on boundary effect correction are included. A complete description can also be found in [99].

5.3.1 Single image deconvolution

The first experiments are based on 256×256 HST images of the planetary nebula NGC7027, galaxy NGC6946 and Crab Nebula NGC19521, with three different integrated magnitudes (m): 10, 12 and 15, not corresponding to the effective magnitudes of these objects but introduced for obtaining simulated images with different noise levels. In Figure 5.5 we show the three objects in the left panels. In the following they will be denoted as Nebula, Galaxy and Crab.

These objects are convolved with an AO-corrected PSF, downloaded from

http://www.mathcs.emory.edu/∼nagy/RestoreTools/index.html, and shown in Figure 5.6. A background of about 13.5 mag arcsec^{-2}, corresponding to observations in K-band, is added to the blurred images and the results are perturbed with Poisson noise and additive Gaussian noise with $\sigma = 10$ e^-/px. According to the approach proposed in Snyder et al. [109], compensation for readout noise is obtained in the deconvolution algorithms by adding the constant $\sigma^2 = 100$ to the images and the background. In Table 5.4 the performances of RL and SGP are reported, in terms of iteration numbers needed to obtain the minimum relative r.m.s. error, CPU times and speedups provided by the two GPU versions with respect to the serial ones. The reconstructions corresponding to minimum relative r.m.s. error, in the case $m = 15$, are shown in the right panels of Figure 5.5.

In the second experiment we use the same datasets created in the previous one to test the effectiveness of the procedure described in Section 2.6 for the reduction of boundary effects. To this aim, the 256×256 blurred and noisy images are partitioned into four partially overlapping 160×160 sub-domains. Each one of the four partial images is merged, by zero-padding, in a 256×256 array which is used, together with the original 256×256 PSF, for the reconstruction of the four parts of the object by means of the RL and SGP algorithms with boundary effect correction. From the four reconstructions 128×128 non-overlapping images are extracted and the complete reconstructed image is formed as a mosaic of them. An example of the result is shown in Figure 5.7. By comparing with the reconstruction of the full image, it is clear that the mosaic of the four reconstruction does not exhibit visible boundary effects.

The results of this experiment for the three objects are reported in Table 5.5. The reconstruction error is the relative r.m.s. error between the mosaic and the original object. By comparing with the results of Table 5.4 we find that the procedure does not increase significantly the reconstruction error. We also point out that we choose the number of iterations corresponding to the *global* minimum, i.e. that providing the best performance on the mosaic of the four reconstructions obtained in the four sub-domains. The computational time is the total time of the four reconstruction.

The third experiment intends to investigate the speedups achievable by SGP when varying the size of the images. We adopt the same procedure already used in Ruggiero et al. (2010). The original 256×256 objects are convolved with

an ideal PSFs and perturbed with background and Poisson noise. Next, images with a larger size are obtained by a Fourier-based re-binning, i.e. the FFT of the original image is expanded by zero padding to a double-sized array and each frequency component is multiplied by 4. In this way the background and the noise level are approximately unchanged. In this experiment we consider only the Nebula and the Galaxy with two magnitudes, 10 and 15. The original images are expanded up to a size of 2048×2048. The results are reported in Tables 5.6 and 5.7, where we highlighted both the speedup observed between GPU and serial implementations (labeled as "Par") and the one provided by the use of SGP instead of RL (labeled as "Alg"). It can be noticed that the computational gain due to the parallel architecture increases proportionally to the size of the image. As far as the speedup of SGP with respect to RL, strong problem-dependent differences in the number of iterations required to reach the minimum errors do not lead to a similarly regular behaviour.

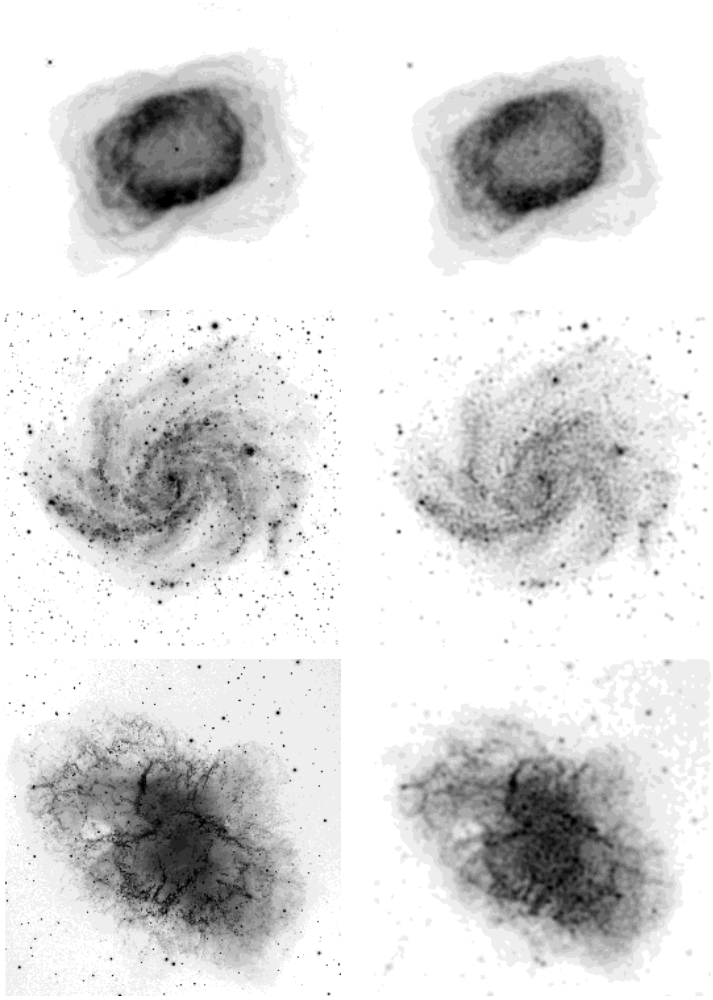


Figure 5.5: The three objects, represented with reverse gray scale (left panels; from up to down Nebula, Galaxy and Crab), and the reconstructions with minimum relative r.m.s. error ($m = 15$; right panels).

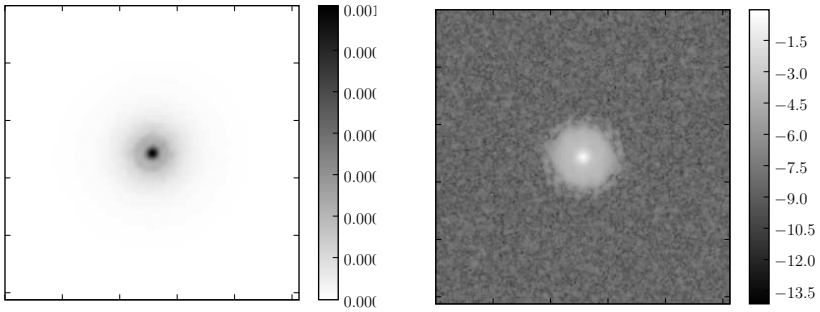


Figure 5.6: The PSF used in the experiments of single image deconvolution (left panel), represented with reverse gray scale, and the corresponding MTF (right panel).

Table 5.4: Iteration numbers, relative r.m.s. errors, computational times and speedups of RL and SGP, provided by the corresponding GPU implementations, for the three 256×256 objects Nebula, Galaxy and Crab. Iterations are stopped at minimum relative r.m.s. error of the serial algorithms.

	Nebula ($m = 10$)				Galaxy ($m = 10$)			
Algorithm	It	Err	Sec	SpUp	It	Err	Sec	SpUp
RL	528	0.021	41.28	-	10000*	0.140	795.3	-
RL_CUDA	528	0.021	2.079	19.9	10000*	0.140	35.09	22.7
SGP	50	0.021	4.719	-	406	0.141	38.61	-
SGP_CUDA	50	0.021	0.344	13.7	406	0.142	3.313	11.7
	Nebula ($m = 12$)				Galaxy ($m = 12$)			
Algorithm	It	Err	Sec	SpUp	It	Err	Sec	SpUp
RL	124	0.026	9.797	-	3887	0.157	304.6	-
RL_CUDA	124	0.026	0.516	19.0	3887	0.157	14.50	21.0
SGP	24	0.026	2.344	-	153	0.159	14.42	-
SGP_CUDA	24	0.026	0.203	11.5	153	0.159	1.266	11.4
	Nebula ($m = 15$)				Galaxy ($m = 15$)			
Algorithm	It	Err	Sec	SpUp	It	Err	Sec	SpUp
RL	124	0.063	9.766	-	448	0.234	35.14	-
RL_CUDA	124	0.063	0.469	20.8	448	0.234	1.594	22.0
SGP	12	0.060	1.250	-	21	0.234	2.094	-
SGP_CUDA	12	0.060	0.109	11.5	21	0.234	0.156	13.4

	Crab ($m = 10$)			
Algorithm	It	Err	Sec	SpUp
RL	5353	0.128	419.8	-
RL_CUDA	5353	0.128	19.45	21.6
SGP	151	0.129	14.28	-
SGP_CUDA	151	0.129	1.219	11.7
	Crab ($m = 12$)			
Algorithm	It	Err	Sec	SpUp
RL	954	0.136	74.83	-
RL_CUDA	954	0.136	3.516	21.3
SGP	52	0.137	4.984	-
SGP_CUDA	52	0.137	0.406	12.3
	Crab ($m = 15$)			
Algorithm	It	Err	Sec	SpUp
RL	128	0.172	10.09	-
RL_CUDA	128	0.172	0.483	20.9
SGP	10	0.172	1.093	-
SGP_CUDA	10	0.172	0.093	11.8

Table 5.5: Reconstruction of Nebula, Galaxy and Crab as a mosaic of the reconstructions of four sub-images with boundary effect correction. The number of iterations is that required for the reconstruction of each sub-domain while the reported computational time is the total time required for the four reconstructions.

	Nebula ($m = 10$)				Galaxy ($m = 10$)			
Algorithm	It	Err	Sec	SpUp	It	Err	Sec	SpUp
RL	818	0.021	243.8	-	10000*	0.144	2813	-
RL_CUDA	818	0.021	12.16	20.0	10000*	0.144	141.5	19.9
SGP	96	0.022	35.16	-	435	0.144	171.6	-
SGP_CUDA	96	0.022	3.406	10.3	435	0.148	14.41	11.9
	Nebula ($m = 12$)				Galaxy ($m = 12$)			
Algorithm	It	Err	Sec	SpUp	It	Err	Sec	SpUp
RL	127	0.026	38.42	-	2347	0.160	696.9	-
RL_CUDA	127	0.026	2.108	18.2	2347	0.160	35.13	19.8
SGP	21	0.026	9.563	-	126	0.161	51.11	-
SGP_CUDA	21	0.026	0.874	10.9	126	0.161	4.438	11.5
	Nebula ($m = 15$)				Galaxy ($m = 15$)			
Algorithm	It	Err	Sec	SpUp	It	Err	Sec	SpUp
RL	96	0.064	27.58	-	297	0.234	89.22	-
RL_CUDA	96	0.064	1.703	16.2	297	0.234	4.547	19.6
SGP	10	0.061	4.234	-	17	0.236	7.375	-
SGP_CUDA	10	0.061	0.407	10.4	17	0.236	0.657	11.2

	Crab ($m = 10$)			
Algorithm	It	Err	Sec	SpUp
RL	4070	0.129	1146	-
RL_CUDA	4070	0.129	61.55	18.6
SGP	129	0.129	46.42	-
SGP_CUDA	129	0.133	4.342	10.7
	Crab ($m = 12$)			
Algorithm	It	Err	Sec	SpUp
RL	696	0.137	196.5	-
RL_CUDA	696	0.137	10.99	17.9
SGP	53	0.137	19.41	-
SGP_CUDA	53	0.137	1.922	10.1
	Crab ($m = 15$)			
Algorithm	It	Err	Sec	SpUp
RL	99	0.172	28.08	-
RL_CUDA	99	0.172	1.704	16.5
SGP	9	0.172	3.859	-
SGP_CUDA	9	0.172	0.360	10.7

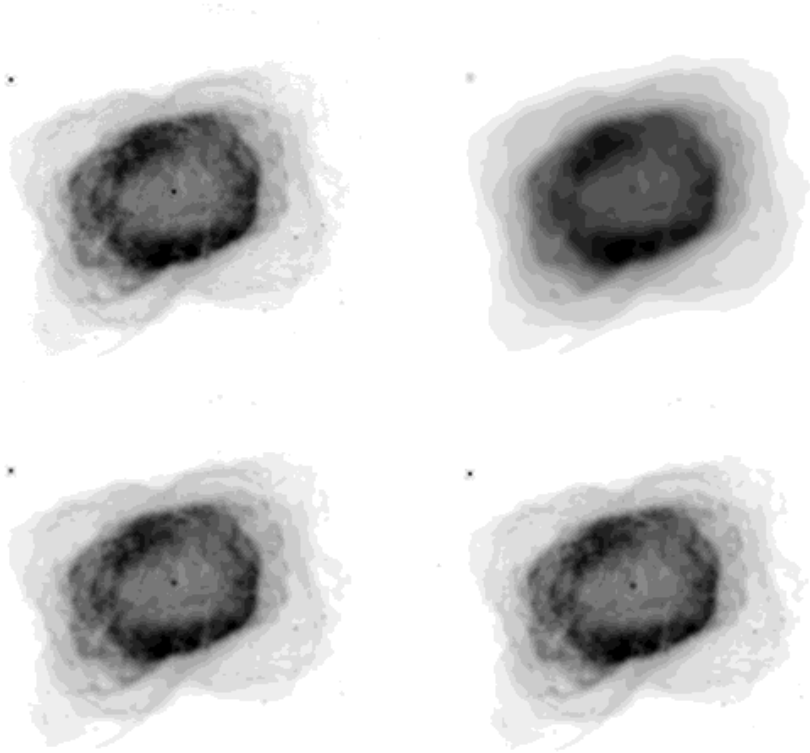


Figure 5.7: Upper-left panel: the original Nebula; upper-right panel: its blurred and noisy image in the case $m = 10$; lower left panel: reconstruction of the global image; lower-left panel: reconstruction as a mosaic of four reconstructions of partially overlapping sub-domains, using the algorithms with boundary effect correction.

Table 5.6: Reconstruction of the Nebula NGC7027 with different image sizes.

$m = 10$					
Algorithm	Size	Err	Sec	SpUp (Par)	SpUp (Alg)
RL It = 10000*	256^2	0.051	783.9	-	-
	512^2	0.051	4527	-	-
	1024^2	0.051	17610	-	-
	2048^2	0.051	80026	-	-
RL_CUDA It = 10000*	256^2	0.051	35.63	22.0	-
	512^2	0.051	69.77	64.9	-
	1024^2	0.051	149.5	118	-
	2048^2	0.051	469.1	171	-
SGP It = 272	256^2	0.052	26.14	-	30.0
	512^2	0.051	143.6	-	31.5
	1024^2	0.051	554.0	-	31.8
	2048^2	0.051	2493	-	32.1
SGP_CUDA It = 272	256^2	0.052	1.797	14.5	19.8
	512^2	0.052	3.469	41.4	20.1
	1024^2	0.052	8.016	69.1	18.7
	2048^2	0.052	25.66	97.2	18.3
$m = 15$					
Algorithm	Size	Err	Sec	SpUp (Par)	SpUp (Alg)
RL It = 612	256^2	0.068	48.27	-	-
	512^2	0.064	278.7	-	-
	1024^2	0.062	1068	-	-
	2048^2	0.062	4897	-	-
RL_CUDA It = 612	256^2	0.068	2.219	21.8	-
	512^2	0.064	4.109	67.8	-
	1024^2	0.062	9.250	115	-
	2048^2	0.062	29.13	168	-
SGP It = 31	256^2	0.068	3.016	-	16.0
	512^2	0.064	16.95	-	16.4
	1024^2	0.062	65.22	-	16.4
	2048^2	0.061	290.8	-	16.8
SGP_CUDA It = 31	256^2	0.068	0.218	13.8	10.2
	512^2	0.064	0.421	40.3	9.76
	1024^2	0.062	1.063	61.4	8.70
	2048^2	0.061	3.406	85.4	8.55

Table 5.7: Reconstruction of the Galaxy NGC6946 with different image sizes.

$m = 10$					
Algorithm	Size	Err	Sec	SpUp (Par)	SpUp (Alg)
RL It = 10000*	256^2	0.293	786.0	-	-
	512^2	0.293	4545	-	-
	1024^2	0.293	17402	-	-
	2048^2	0.293	80022	-	-
RL_CUDA It = 10000*	256^2	0.293	36.64	21.5	-
	512^2	0.293	67.94	66.9	-
	1024^2	0.293	146.7	119	-
	2048^2	0.293	463.9	172	-
SGP It = 928	256^2	0.292	88.72	-	8.86
	512^2	0.291	484.3	-	9.38
	1024^2	0.291	1854	-	9.19
	2048^2	0.291	8386	-	9.54
SGP_CUDA It = 928	256^2	0.293	7.219	12.3	5.08
	512^2	0.293	11.14	43.5	6.10
	1024^2	0.293	25.86	71.7	5.67
	2048^2	0.293	81.02	104	5.73
$m = 15$					
Algorithm	Size	Err	Sec	SpUp (Par)	SpUp (Alg)
RL It = 1461	256^2	0.311	114.9	-	-
	512^2	0.307	644.3	-	-
	1024^2	0.306	2574	-	-
	2048^2	0.306	11689	-	-
RL_CUDA It = 1461	256^2	0.311	5.375	21.4	-
	512^2	0.307	9.656	66.7	-
	1024^2	0.306	22.41	115	-
	2048^2	0.306	68.44	171	-
SGP It = 38	256^2	0.311	3.672	-	31.3
	512^2	0.308	20.36	-	31.6
	1024^2	0.307	78.20	-	32.9
	2048^2	0.306	354.0	-	33.0
SGP_CUDA It = 38	256^2	0.311	0.266	13.8	20.2
	512^2	0.307	0.531	38.3	18.2
	1024^2	0.307	1.344	58.2	16.7
	2048^2	0.306	4.188	84.5	16.3

5.3.2 Multiple images deconvolution

In this Section we test the efficiency of three algorithms for multiple image deconvolution, i.e. multiple RL, OSEM and SGP (applied to Multiple RL), by means of simulated images of the Fizeau interferometer LINC-NIRVANA (LN, for short; T. Herbst et al., [71]) of the Large Binocular Telescope (LBT).

In our simulations we use PSFs generated with the code LOST (Arcidiacono et al. [6]); one of them, with SR = 70% and horizontal baseline, is shown in Figure 4 together with the corresponding MTF. Moreover, we consider two test objects: one is again the Nebula NGC7027, with two magnitudes, 10 and 15, and size 512×512 (therefore, the images are noisier than those of the 256×256 version with the same integrated magnitude); the other is a model of open star cluster derived from an image of the Pleiades (Star Cluster, for short), consisting of 9 stars with magnitudes ranging from 12.86 to 15.64. These objects are convolved with three PSFs corresponding to three equispaced orientations of the baseline, 0°, 60°, and 120°. In the *u,v* plane they provide a satisfactory coverage of the band of a 22.8m telescope (see, for instance, Bertero et al. [16], a review paper where the generation of the images used in these tests is described with greater detail). The results are perturbed with a background of about 13.5 mag arcsec^{-2}, corresponding to observations in K-band, and with Poisson and Gaussian noise ($\sigma = 10\ e^-$/px). In Figure 5 we show one interferometric image of the Nebula, with magnitude 15, and one interferometric image of the Star Cluster, both with horizontal baseline.

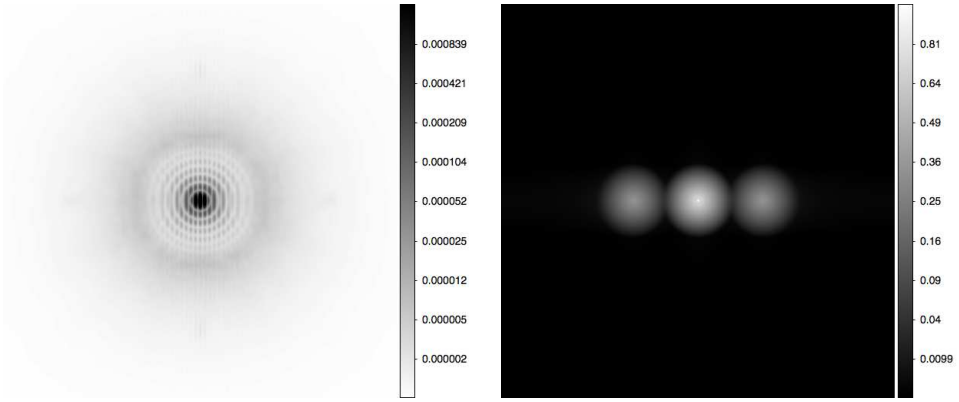


Figure 5.8: Simulated PSF of LINC-NIRVANA with SR = 70% (left panel) and the corresponding MTF (right panel). The PSF is monochromatic in K-band and is the PSF of a 8.4m mirror (the diameter of the two mirrors of LBT) modulated by the interferometric fringes. Accordingly, in the MTF the central disk corresponds to the band of a 8.4m mirror while the two side disks are replicas due to interferometry.

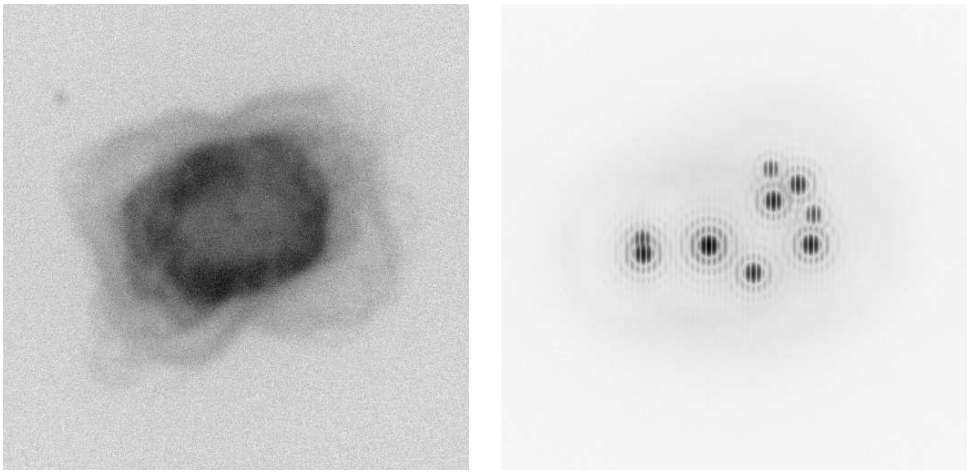


Figure 5.9: Interferometric images (horizontal baseline) of the 512×512 Nebula with $m = 15$ (left panel) and of the star cluster (right panel).

Diffuse objects In this Section we give the results obtained in the case of the Nebula with two magnitudes, 10 and 15. The stopping rule is given again by the minimum r.m.s. error. We first consider deconvolution without correction for edge artifacts because the object is interior to the image domain. The results are reported in Table 5.8. If we compare the behaviors of single image and multiple image RL, we find that in the second case a larger number of iterations is required, due to the difficulty of combining the resolutions of the different images to get a unique high-resolution reconstruction. Moreover, the greater cost per iteration is due to two causes: the first is that the size is 256×256 in the single case and 512×512 in the multiple image case; the second is that one single image iteration requires 4 FFTs, while one multiple image iteration, with three images, requires 10 FFTs.

The results confirm that the speedup provided by OSEM with respect to multiple RL is about 2.5 with a reduction by a factor 3 of the number of iteration (see Section 2.3) but the speedup provided by SGP with respect to OSEM is interesting, by a factor between 6 and 7. Presumably this speedup decreases with increasing number of images, but a speedup of about 20 is provided by OSEM in the case of 26 images, a number that presumably will never be reached in the case of LN. Therefore one can conclude that SGP can be recommended for the deconvolution of LN images. CUDA implementations provide an additional speedup of about 80/90 for RL and OSEM, while smaller factors are observed for SGP.

For testing the accuracy of the deconvolution methods with boundary effect correction we follow the same procedure used in the single image case, i.e. the images are partitioned into 4 partially overlapping sub-images, the methods with boundary effect correction are applied and the final reconstruction is obtained as a mosaic of the four partial reconstructions. The results are reported in Table 5.9 and confirm the results obtained in the single image case.

Table 5.8: Reconstruction of the Nebula using three equispaced 512×512 images.

$m = 10$				
Algorithm	It	Err	Sec	SpUp
RL	3401	0.032	4364	-
RL_CUDA	3401	0.032	48.00	90.9
OSEM	1133	0.032	1602	-
OSEM_CUDA	1133	0.032	18.59	86.2
SGP	144	0.033	220.7	-
SGP_CUDA	144	0.033	3.563	61.9
$m = 15$				
Algorithm	It	Err	Sec	SpUp
RL	353	0.091	441.5	-
RL_CUDA	353	0.091	4.937	89.4
OSEM	117	0.091	165.7	-
OSEM_CUDA	117	0.091	2.062	80.4
SGP	16	0.087	26.14	-
SGP_CUDA	16	0.087	0.546	47.9

Table 5.9: Reconstruction of the Nebula as a mosaic of 4 reconstructed sub-images with boundary effect correction, also in the case of three equispaced images.

$m = 10$				
Algorithm	It	Err	Sec	SpUp
RL	2899	0.034	13978	-
RL_CUDA	2899	0.034	174.2	80.2
OSEM	950	0.034	5447	-
OSEM_CUDA	950	0.034	64.03	85.1
SGP	160	0.034	873.3	-
SGP_CUDA	160	0.034	15.45	56.5
$m = 15$				
Algorithm	It	Err	Sec	SpUp
RL	243	0.094	1174	-
RL_CUDA	243	0.094	15.28	76.8
OSEM	81	0.094	479.1	-
OSEM_CUDA	81	0.094	5.939	80.7
SGP	11	0.087	69.88	-
SGP_CUDA	11	0.086	1.532	45.6

5.3.3 Point-wise objects

In this case iterations are pushed to convergence and therefore the stopping rule is given by condition (3.1); we use different values of *tol*, and precisely 10^{-3}, 10^{-5} and 10^{-7}. In order to measure the quality of the reconstruction we introduce an average relative error of the magnitudes defined by

$$\text{av\_rel\_er} = \frac{1}{q}\sum_{j=1}^{q}\frac{|m_j - \widetilde{m}_j|}{\widetilde{m}_j} \quad , \tag{5.1}$$

where q is the number of stars (in our case $q = 9$) and $\widetilde{m}_j$ and m_j are respectively the true and the reconstructed magnitudes. The results are reported in Table 5.10.

First we point out that, as in the previous cases, we constrain the parallel codes to perform the same number of iterations as the serial ones. This constraint is introduced because the FFT does not have the same precision in the two cases, as already discussed. As a result, the two implementations of the same algorithm do not provide the same error with the same number of iterations. This effect presumably will be removed when a double-precision FFT will be available for GPU in GPUlib.

Next, we find, as expected, that the number of iterations increases with decreasing values of *tol*. But the increase in computation time is not compensated by a significant decrease in the accuracy of the reconstructed magnitudes. With $tol = 10^{-3}$ the accuracy of the estimated magnitudes could already be satisfactory. We observe however that with this milder tolerance the accuracy provided by the three algorithms is not the same. Multiple RL and OSEM seem to be slightly more accurate. The accuracy of all algorithms is essentially the same for the smaller tolerances.

As a final experiment we consider the reconstruction of a binary with high dynamic range (Bertero et al. [16]). It consists of a primary with $m_1 = 10$ (denoted as S_1) and a secondary with $m_2 = 20$ (denoted as S_2). The distance between the two stars is 45ms (i.e. 9 pixels for the LINC-NIRVANA detector) and the axis of the binary forms an angle of 23° with the direction of the baseline of the first image. Three equispaced images are generated as in the case of the Star Cluster, using the same PSFs and the same background.

In this experiments we need a very small tolerance, i.e. $tol = 10^{-7}$, in order

Table 5.10: Reconstruction of the Star cluster with three 512×512 equispaced images. The error is average relative error of the magnitudes defined in Eq. (5.1).

	tol = 1e-3			
Algorithm	It	Err	Sec	SpUp
RL	319	2.39e-4	393.4	-
RL_CUDA	319	2.38e-4	4.641	84.8
OSEM	151	1.63e-4	220.8	-
OSEM_CUDA	151	1.62e-4	2.421	91.2
SGP	71	1.35e-3	97.80	-
SGP_CUDA	71	1.29e-3	1.641	59.6
	tol = 1e-5			
Algorithm	It	Err	Sec	SpUp
RL	1385	6.65e-5	1703	-
RL_CUDA	1385	6.64e-5	19.38	87.9
OSEM	675	5.64e-5	980.6	-
OSEM_CUDA	675	5.64e-5	10.75	91.2
SGP	337	5.89e-4	455.2	-
SGP_CUDA	337	1.79e-4	7.187	63.3
	tol = 1e-7			
Algorithm	It	Err	Sec	SpUp
RL	7472	5.64e-5	9180	-
RL_CUDA	7472	5.98e-5	104.8	87.6
OSEM	3750	6.13e-5	5442	-
OSEM_CUDA	3750	5.98e-5	59.52	91.4
SGP	572	7.37e-5	772.6	-
SGP_CUDA	572	7.05e-5	12.20	63.3

to allow to SGP to detect the faint secondary. The reason is presumably that SGP requires a projection on the non-negative orthant and the existence of this projection can make difficult the appearance of the secondary. Anyway, the results reported in Table 5.11 are interesting and demonstrate that also the magnitude of the secondary can be estimated with a sufficient accuracy and a reasonable computation time.

5.3.4 Discussion

The results given in the previous Sections demonstrate that SGP allows a significant speedup of all the RL-type algorithms considered.

Table 5.11: Reconstruction of the Binary with high dynamic range (image size: 256×256).

	tol = 1e-7		
Algorithm	It	Sec	SpUp
RL	30765	6108	-
RL_CUDA	30765	292.9	20.9
OSEM	14291	3216	-
OSEM_CUDA	14291	156.0	20.6
SGP	2073	482.8	-
SGP_CUDA	2073	28.59	16.9
	Magnitude		
Algorithm	Star	Real	Reconstructed
RL	S1	10	10.0001
	S2	20	20.1841
OSEM	S1	10	10.0001
	S2	20	20.0919
SGP	S1	10	10.0001
	S2	20	20.2683

In Section 5.2 we presented a computational comparison of different parallel implementations on various types of HPC parallel architectures. Hence, we show how a careful MPI-based parallel implementation of the SGP method for very recent concurrent architectures allows to efficiently face large- and huge-scale image reconstruction problems in reasonable time. Moreover, we compare these results against a CUDA-based parallel implementation for GPU architectures: even if it has been shown how effective can be the SGP-based approach on these machines, we show that for large-scale imaging problems the MPI-based version allows to overcome the memory limitations of GPUs, thus providing a very powerful tool to solve real-world multidimensional problems.

Here (5.3) can be noticed that the speedup ranges from about 4 in the case of multiple images of the Star Cluster (Table 5.10), to more than 30, in the case of a single image of the Galaxy (Table 5.7). A more accurate investigation of the speedup achievable should require application to a broader data set of astronomical objects as well as to images with different noise levels and noise realizations. Anyway we believe that the results presented are sufficient to demonstrate that SGP is a valuable acceleration of RL-like algorithms and that in several cases it

allows a considerable reduction in computational time.

The speedup provided by GPU implementation is consistent with the results reported in Ruggiero et al. [103]. The speedup of RL-algorithms is greater than that of SGP-algorithms because the main computational kernel of RL is FFT while SGP requires also the computation of steplengths etc. Anyway, the gain with respect to RL is still very significant. In some cases, it allows to deconvolve a 2048×2048 image in a few seconds.

5.4 Microscopy: 3D image restoration

Finally, to reach also the microscopy community, where large-scale imaging problems are abundant, we want to provide them quasi real-time deconvolution algorithms able to drastically reduce the time for the pipe-line image analysis. We used both CLSM and STED microscopy images to demonstrate the speedup of the SGP based algorithms. However, the very same algorithms can be applied to any other fluorescence microscopy technique by simply providing the relative point spread function, or more generally the relative forward model.

We test the proposed algorithms both on synthetic and real data of confocal and STED microscopy and results have been shown in [125] Combining the SGP method with the GPU implementation we achieve a speedup factor from about a factor 25 to 690 (with respect the conventional algorithm). The excellent results obtained on STED microscopy images demonstrate the synergy between super-resolution techniques and image-deconvolution. Further, the real-time processing allows conserving one of the most important property of STED microscopy, i.e the ability to provide fast sub-diffraction resolution recordings.

Our test platform consists of a workstation equipped with 2 Intel Xeon SixCore CPUs at 3.1GHz, 188GB of RAM and 4 GPUs Nvidia Fermi C2070. It is managed by a CentOS Linux distribution. Each GPU is highly parallel: 14 streaming multiprocessors for a total of 448 64bit computing cores, a high-speed RAM block shared among the 448 cores and a cache.

Two implementations of SGP are available: one in Matlab (CPU-based) and another one in C/CUDA (GPU-based). Here we used Matlab v. 7.11 and CUDA v. 4.3.

5.4.1 Synthetic images

To generate pseudo-random phantoms which mimic the micro-tubule network of a cell we randomly selected the starting positions of a given and fixed number of filaments and successively we used a stochastic process for choosing iteratively the directions of growth. The growth has been performed in a bi-dimensional or three-dimensional space to obtain 2D or 3D phantom, respectively. We assumed filaments having tubular structure with radius 30 nm and we introduced heterogeneity of protein concentration between different filaments by associating to each filament a value in the range $[0, 1]$. Successively, to obtain the ideal image we convolved the phantom $\boldsymbol{x}$ with the system PSF, *i.e.*, $A\boldsymbol{x}$. Importantly, the PSF of the STED system becomes narrower respect to the confocal counterpart as the saturation factor ς increases, but the intensity value at the peak stays constant. Thereby, in the convolution process, we used Equations (1.19) and (1.20) without any normalization to the sum of the pixels/voxels.

To obtain the ideal image in terms of average detected photons, we multiplied the convolved object by a factor τ which depends on several multiplicative factors, such as the emission rate of the fluorophore, the collection efficiency of the system and the pixel dwell-time. Since we assumed that photon counting noise represents the major source of noise for the detection process, and a constant background b can further degrade the image, we obtained the final image by corrupting every pixel/voxel i with a Poisson process with mean $\tau(A\boldsymbol{x})_i + b$. Thus by increasing τ, the average number of detected photons increases and hence the signal-to-noise ratio (SNR) increases. The relation between SNR and τ is

$$SNR_{db} = 10 \log \left(\max_i \frac{\tau(Ax)_i}{\sqrt{\tau(Ax)_i + b}} \right) \quad . \tag{5.2}$$

Since in simulation we know the object $\boldsymbol{x}$ used to generated the simulated image, we can numerically evaluate the quality of the deconvolved images at each iteration k. Notably, we computed the KL divergence by using the phantom $\boldsymbol{x}$ scaled with the effective photons emitted from each pixel/voxel. In the case of simulated data, we stopped the algorithms when they reached the minimum of the KL divergence.

5.4.2 Real images

To test the proposed algorithms we imaged the micro-tubule cytoskeleton of fixed PtK2 cells. We used two different protocols for labeling two different proteins of the filament systems. The first protocol localizes the β-tubulin protein; it involves a primary antibody (antiβ-tubulin mouse IgG, Sigma) and a secondary antibody (sheep anti-mouse IgG, Dianova) labeled with ATTO 647N (Atto-Tec). The second protocol localizes keratin protein and uses the Citrine yellow fluorescent protein. The samples were examined using a Leica TCS STED-CW microscope (Leica Microsystems) equipped with a 100x/1.40 OIL STED orange objective (Leica Microsystems). The system is able to perform both confocal and STED imaging. We excited (with a regular Gaussian beam) ATTO 647N and Citrine fluorophores at 635 nm and 488 nm, respectively, and we collected emitted light in the 650-750 nm and 495-580 nm spectral windows, respectively. For STED imaging on Citrine tagged sample fluorescence we depleted with a doughnut shaped beam at 592 nm. In the case of real data, we stopped the algorithms when they reached convergence with a tolerance of 10^{-3}.

We first used bi-dimensional (2D) CLSM and STED microscopy synthetic images for comparing the well-known RL algorithm with the SGP-based algorithm (both minimizing the functional described in equation (1.11)). Realistic phantoms are crucial when robustness of algorithms has to be evaluated. For this reason, we implemented a routine able to generate pseudo-randomly phantoms which mimic the micro-tubule cytoskeleton of a cell. We simulated the images of the two microscopy modalities by using the same random tubulin network specimen (Figure 5.10a) and the same imaging conditions, but two different point-spread-functions (PSFs) (Insets Figure 5.10b) for mimicking the different spatial resolutions. In particular, we assumed a Gaussian shaped PSF with a full-width-half-maximum (FWHM) of 220 nm for CLSM ($\sigma_r = 93$ nm) and a Gaussian-Lorentzian shaped PSF with a FWHM of 100 nm for STED microscopy ($\sigma_r = 93$ nm, $\psi = 3.22{\cdot}10^{-3}$ nm^{-1}, $\varsigma = 7$). Figure 5.10e-l shows a side-by-side comparison of RL- and SGP-based restorations. Clearly, the STED image reveals superior details compared to the CLSM image because their differences in spatial resolution (Figure 5.10b,c,d,Figure 5.15). Importantly, we deconvolved the synthetic images by means of the very same PSFs used for their generation (inverse crime). Thereby the results were not fundamentally biased by the choice of the

PSF. After deconvolution we obtained excellent contrast improvement and noise reduction that help distinguishing more structural details into the CLSM restored images (Figure 5.10e,g,i,k), as well as, into the STED microscopy restored images (Figure 5.10f,h,j,l).

More interesting for the scope of this work is the comparison between RL- and SGP-based restorations. Both algorithms led to similar results (Figure 5.10e,f,i,j). However close looks to the restorations (Figure 5.10h,l,g,k) depict slight differences. For example the SGP-based images offers higher contrast with respect to the RL-based counterpart. A quantitatively analysis confirmed such improvement (Figure 5.15). Even if RL and SGP algorithms converge to the same minimum, they follow different approximation paths and the restorations satisfying the stopping rule can present marginal differences.

Whereas RL and SGP algorithms are similar in terms of restoration quality they have strong differences in terms of speed. Figure 5.11 plots the time and the number of iterations requested for obtaining optimal (in terms of restoration accuracy) restored images as function of the image size (number of pixels). We confirmed robustness of the algorithms against noise and object structures by running the algorithms with different noise realizations and different random tubulin network realizations. Moreover, we carefully maintained the concentration of filaments constant for all image sizes, in order to remove any dependency of the iterations' optimal number on the image size. On the other side, the optimal number of iteration changes between the two microscopy techniques. However, the reason is not connected to the microscopy technique itself, but to the different intensity-dynamic of their images (Color bars Figure 5.10a) and the different size of the their PSFs. As a rule of thumb the optimal number of iterations increases for increasing intensity-dynamic (number of photons collected *per* pixel) and blurring of the image (size of the PSF with respect to the pixel size).

More interesting for SGP and RL algorithms comparison is that SGP reduces the optimal number of iterations ($\sim$ 87% for CLSM and $\sim$ 51% for STED microscopy). This is in agreement with the main feature of the SGP method, *i.e.*, the ability to find optimal direction toward the minimum of the functional and thereby to reduce the number of iterations needed. However, a fair comparison of the speedup of SGP algorithm has to take into account that a single SGP iteration needs more computation than a single RL iteration. Thereby, the overall time

speedup obtained with the SGP algorithm is $\sim 20\%$ for STED microscopy and $\sim 80\%$ for CLSM. Similarly to the RL algorithm, also the SGP algorithm decreases the optimal number of iterations when the signal-to-noise ratio (SNR) decreases. Thus, in a regime of very low SNR the speedup of the SGP-based algorithm with respect to the RL algorithm can reduce (Figure 5.16).

After estimating the speedup related to the SGP algorithm alone, we evaluated the further speedup obtained by implementing the SGP algorithm for GPU (instead of CPU). Figure 5.12 shows the time needed to obtain optimal restoration as a function of the image size. The GPU-based algorithm works ~ 10 times faster for small images (126×126 pixels) and ~ 100 times faster for large images (4096×4096 pixels) when compared to the CPU-based algorithm. Notably, this speedup has to be added to the speedup provided by the SGP algorithm, for example for large CLSM images (4096×4096 pixels) the GPU-based SGP algorithm need ~ 8s, that is ~ 690 times faster then the CPU-based RL algorithm. If we consider that to achieve adequate SNR a modern CLSM need a pixel-dwell time of at least 1 μs, in this example the deconvolution process is at least 2 times faster then the time to collect the image.

Next, motivated by the promising results on synthetic images we applied the SGP algorithm to real images of tubulin network (Figure 5.13). In contrast to results on synthetic images, results on real images strictly depend on the PSF. Thereby, even if any method which estimate the PSF is fully compatible with the proposed algorithms, one has to pay particular attention to the PSF choice. A PSF may be empirical, *i.e.*, measured [107] or theoretical, *i.e.*, calculated [55]. Empirical PSF is generally obtained by imaging of sub-resolved structures in the same system conditions (i.e optics and specimen's environment) used to image the specimen. Whereas calculated PSF is generated by using analytical models which require parameters like wavelength configurations, objective lens details, refractive indexes of immersion and mounting media, etc. Both methods present advantages and disadvantages. Briefly, an empirical PSF is contaminated by noise and has to be measured exactly in the same conditions that will be used to image the specimen, on the other side, a measured PSF takes into account any kind of aberration that can arise in the whole system, including aberration introduced by the specimen itself; a theoretical PSF is noise-free, but, its computation requires many information that are not easy to known and complex models. Also, a third option

exists where the PSF is estimated from the image together with the unknown object, *i.e.*, blind deconvolution [72]. In this case we adopted an hybrid method: we used a rather easy PSF parametric model whose parameters are directly extracted from the image of sub-resolved structures contained in the very same specimen ($\sigma_r = 93$ nm, $\psi = 3.22{\cdot}10^{-3}$ nm^{-1}, $\varsigma = 5.2$). Importantly, in the case of STED deconvolution is extremely important to estimate the PSF directly from the image being deconvolved since the PSF strictly depends also by the property of the fluorescent marker. For example, the use of fluorescent beads can result in a wrong estimation of the PSF, since in most of the case the fluorescent marker used for the beads is different from the one used for labeling the specimen.

The superior resolution of STED microscopy clearly highlights filaments intersection that can not be resolved in the CLSM counterpart (Figure 5.13a-d). By strongly improving the contrast and reducing the noise, the SGP algorithm is able to recover many structural details from the raw CLSM image, as well as from the raw STED microscopy images. These results fully confirm the importance of applying deconvolution also to super-resolution techniques, such as STED microscopy. Moreover, this example clarifies which are the benefits of using algorithms based on equation (1.11), like RL and SGP. It is well known that minimization of equation (1.11) leads to pointwise (sparse) restorations. For this reason many regularization methods have been proposed by different groups in order to apply deconvolution also for imaging of piecewise structures. In this study we applied deconvolution on tubulin network images, which is a rather sparse structure. SGP algorithm offers superior results when it is applied to reconstruct single isolated tubulin filaments. There are almost no differences between CLSM and STED microscopy restoration when comparing the intensity profile through a single isolated filaments (Figure 5.13j), *i.e.*, deconvolution on CLSM can, in these particular circumstances, substitutes STED microscopy. On the contrary, when more convoluted structures are imaged, the lower resolution offered by CLSM microscopy can not be compensated by deconvolution. STED microscopy, especially when combined with deconvolution, easily resolves two close (<100 nm) tubulin filaments (Figure 5.13i), but CLSM, even if combined with deconvolution, fails on the same task (Figure 5.13i). The GPU-based SGP algorithm provided the restoration in ~ 0.07 s (21 iterations) and ~ 0.16 s (45 iterations) for CLSM and STED microscopy, respectively. Indeed, ~ 37 (STED) and ~ 16 (CLSM) time

faster than the time that the microscope need to produce the images. The advantages of using the GPU-based algorithm becomes plain for 3D data set. We tested the SGP algorithm on a 3D CLSM image of the entire cytoskeleton of a cell (Figure 5.14a) which took 180 s to be collected. Despite the huge data set ($1024 \times 1024 \times 33$ voxel) the GPU-based implementation of the SGP algorithm obtained an excellent restoration (Figure 5.14b) after 20 iterations taking $\sim$ 35s, which is about a factor ~ 5 and ~ 35 faster than the collection time and the time need by the CPU-based implementation, respectively. Finally, we remark that we obtained all the results working with double precision, thereby a further reduction of running time is expected when using single precision. For example, when working in single precision the entire cytoskeleton 3D restoration needed $\sim$ 17 s, thereby ~ 2 time faster. Importantly, we observe that in the microscopy contest running the deconvolution algorithms in single and double precision we obtained similar qualitative results.

5.4.3 Discussion

In this Section 5.4, we have shown that image deconvolution can potentially improve image quality for any fluorescence microscopy technique, including the new emerging nanoscopy techniques. However, the amount of computational time required, which characterizes any high performance algorithm, has so far limited the massive spreading of image deconvolution. Thus we presented a framework able to efficiently reduce the computational time for solving both the ML (un-regularized) and the MAP (regularized) deconvolution problem. This framework uses the SGP method for solving the minimization problem associated to deconvolution. As an example, we use this framework to derive an efficient alternative to one of the most used deconvolution algorithm in fluorescence microscopy, the RL algorithm. Furthermore, we compared CPU-based and GPU-based implementations of this algorithm. The synergy between the SGP method and the GPU-based implementation achieves an improvement which ranges from about a factor of 25 to 690 (when compared to a CPU-based implementation of the RL algorithm), without loosing in quality of the reconstruction.

It is important however to point out the limitations of the SGP method which, in this work, is mainly applied to the ML problem because we are focusing on possible real-time applications. As previously remarked, the SGP method can also be

applied to the solution of regularized problems (and one example is provided), but only if the regularization function is differentiable. This is an important limitation because, in general, the SGP method can not be applied to the important case of sparse reconstruction schemes, i.e. ℓ_1-norm regularization. More precisely, it can be applied to the case of edge-preserving regularization if a smoothed TV-norm is used [124], but not to the case of sparsity of the object with respect to a suitable wavelet transform, such as a dual-tree complex wavelet transform or a dictionary composed of curvelets and un-decimated wavelet transform [49, 27], an approach already proposed for confocal microscopy.

In the case of a piece-wise object with sharp edges, regularization by early stopping of un-regularized SGP or RL can produce a smoothing of the edges and therefore edge-preserving regularization is required. This over-smoothing effect does not appear in the restoration of tubulines networks because, as we already remarked, this is essentially a sparse object and the ML solutions are sparse in the pixel space.

In the case of regularized methods an important point is the choice of the regularization parameter. For any selection criterion the solution of several minimization problems is in general required so that a real-time application is not possible. A way could be the approach proposed in [27], where the choice of the parameter is reduced to the solution of a unique constrained minimization problem with an additional constraint related to the selection criterion. We believe that also this constrained minimization problem is too much time-consuming to enable real-time deconvolution with the available GPU technology. A more practical way could be to calibrate off-line the regularization parameter for a given class of objects (for instance tubulines) and a given value of the signal-to-noise ratio. Then the estimated value could be used for real-time SGP-based deconvolution.

The question if conventional microscopy combined with image deconvolution alone (without using prior information about the object to reconstruct) can recover object's frequencies beyond the cut-off frequency of the system (i.e. achieve sub-diffraction resolution) is still controversial. In the case of STED microscopy the situation is rather different. In a STED microscope the response of the object's emission rate to the illumination is nonlinear (exponential). Roughly speaking, this property allows to the STED microscope system to transfer all the object's frequencies (no cut-off frequency exists), thus permitting theoretically unlimited

resolution [68, 118]. However, the strength of the frequencies declines rapidly with the increases of the order, leaving the practical resolution finite due to signal-to-noise concerns. On the other side, in a STED microscope the strength of the high frequency can be enhanced by increasing the intensity of the illumination, which unfortunately can also introduce photodamage effects on the specimen. In this scenario, image deconvolution can efficiently help recovering high frequencies which are transmitted by the microscope system but hindered by the noise, thereby improving the practical resolution without increasing the intensity of the illumination.

Motivated by the good results we obtained, it is easy to think that further developments can be considered in the way of hybrid programming, that is by mixing inter-node distributed-memory computations (the MPI-related part) with intra-node shared-memory multithreaded computations (the OpenMP-related part). Here the computing nodes are thought to be multicore CPUs. This is an higher level of parallelization, which in recent years has shown to be very effective in catching the benefits of both the programming paradigms. A second line of research is represented by the integration of GPUs and multicore CPUs: these configurations are increasingly appealing for their reduced costs and good performances. Servers equipped with a bunch of multicore CPUs and a limited number of last-generation GPUs devices are affordable HPC architectures, able to reach Tflops-level performances. The implementation of the SGP approach within such a mixed environment is surely possible, but has a number of nontrivial issues to face. Nevertheless, it would make possible to solve huge image restoration problems in a very limited time, thus opening the way of HPC optimization to a lot of meaningful applications in many fields.

Work is in progress for developing a library of SGP algorithms for Poisson data deconvolution with a number of different kinds of regularization.

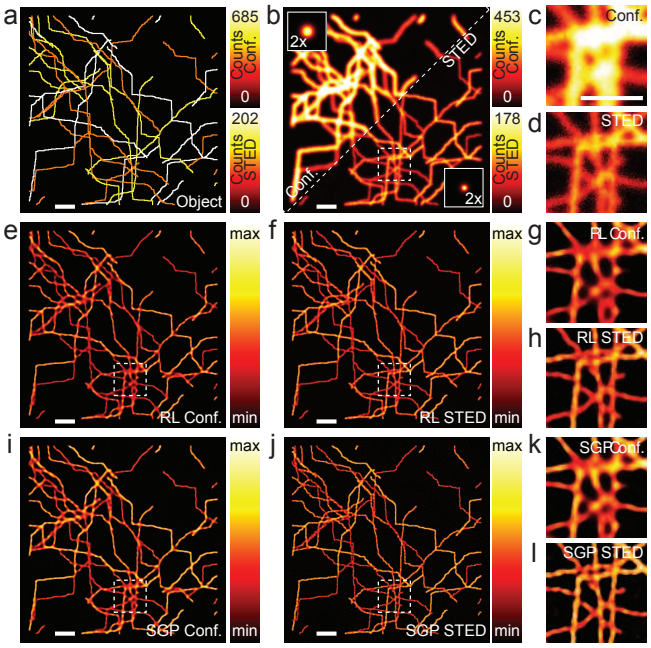


Figure 5.10: **Comparison between SGP- and RL-based restorations of 2D synthetic data.** (a) Phantom mimicking the micro-tubulin cytoskeleton of a cell (512×512 pixels, 20 nm pixel size). In particular the map depicts the distribution of fluorescent photons being emitted by each pixel. The two color bars represent respectively the CLSM and STED microscopy case. Since STED microscopy improves the resolution by reducing the effective area from which fluorescence is emitted, the amount of photons in the STED microscopy case reduces. (b) Synthetic images for the CLSM (upper left, SNR $\sim$ 13 db) and STED microscope (lower right, SNR $\sim$ 11 db). Insets represent the respective PSFs magnified by a factor 2. (c,d) Magnified views of the area denoted by the dashed box in (b) for CLSM and STED microscopy, respectively. (e,f,i,j) RL-based (e,f) and SGP-based (i,j) restorations of CLSM (e,i) and STED (f,j) images. (g,h,k,l) Magnified views of the area denoted by the white box in (e,f,i,j). All the magnified views are renormalized in signal intensity. Algorithms were stopped using the minimization of the KL-divergence as criterium). Scale bars: 1 μm.

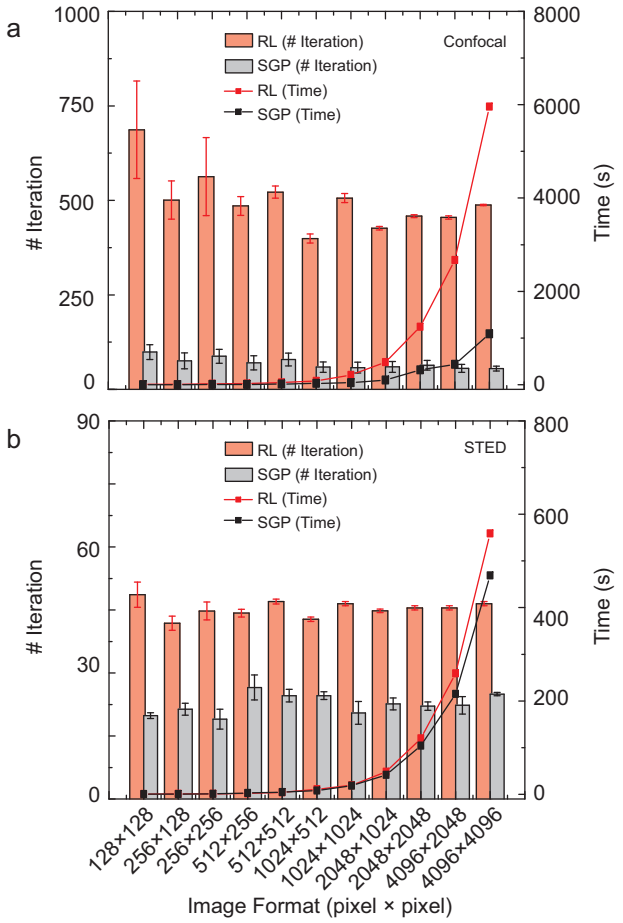


Figure 5.11: **Performance comparison between the SGP and RL algorithms.** (a,b) Number of iterations and computational time as a function of image size for CLSM (a) and STED microscopy (b). All the test compared the CPU-based implementations. Each point represents the mean and the standard deviation of 20 different realizations, in particular 10 different noise realizations for 2 different phantom realizations. Same conditions of Figure 5.10.

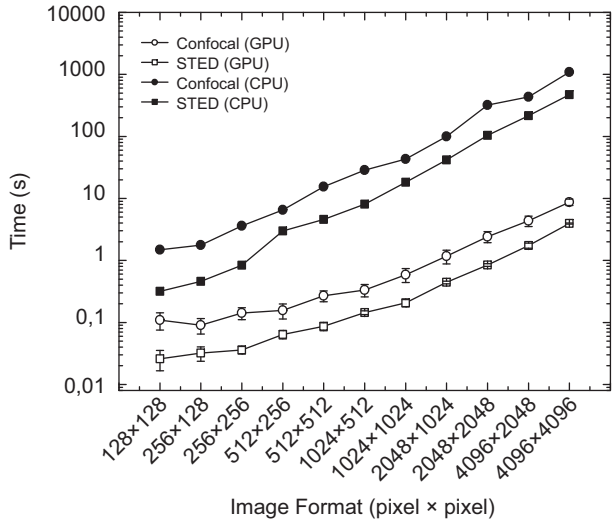


Figure 5.12: **Performance comparison between CPU- and GPU-based implementation of the SGP algorithm.** Computational time as a function of image size. Same conditions of Figures 5.10 and 5.11.

.

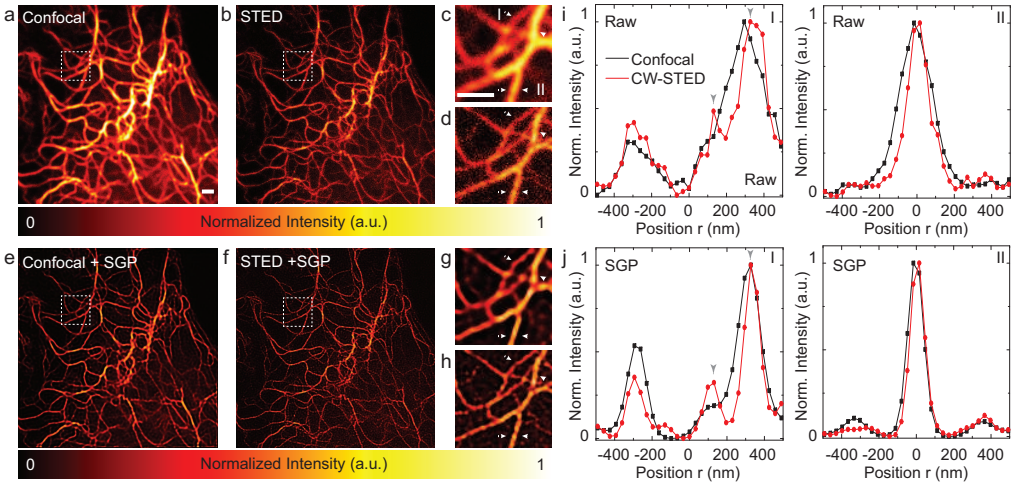


Figure 5.13: **SGP-based restoration of 2D real CLSM and STED microscopy images.** (a,b) Bidimensional CLSM (a) and STED microscopy (b) images of the micro-tubulin network of a PtK2 cell (512 × 512 pixel, pixel dwell-time 10 μs, pixel size 32 nm). (c,d) Magnified views of the area denoted by the dashed box in (a) and (b), respectively. (e,f) SGP-based restored image of (a) and (f), respectively. (g,h) Magnified views of the area denoted by the white box in (e) and (f), respectively. (i,j) Intensity profiles (along the lines between the arrows) through single isolated filament (I) or close-packed filaments (II) in the raw (c,d) and restored (g,h) images. Scale bars: 1 μm.

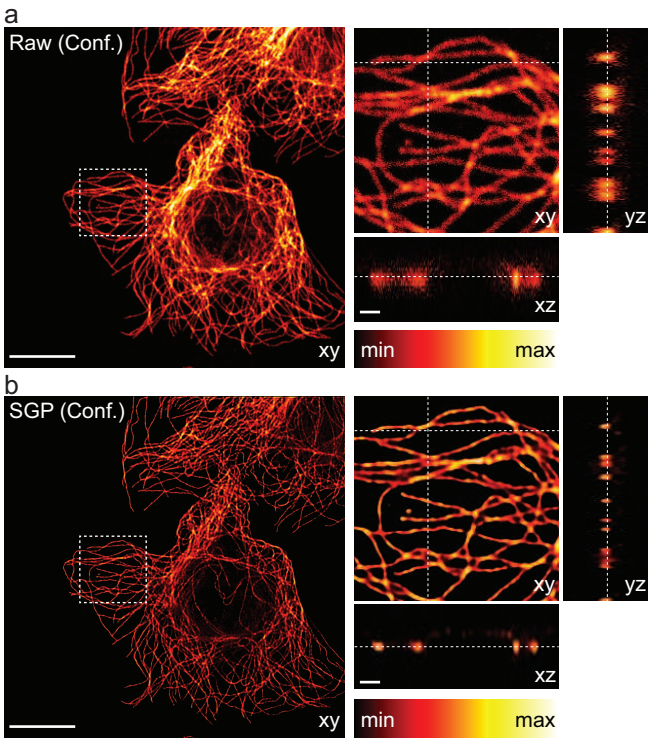


Figure 5.14: **SGP-based restoration of 3D large real CLSM data set**. (a) Middle xy section of the raw data (1024 × 1024 × 33 pixel, pixel dwell-time 5 μs, pixel size 50 × 50 × 125 nm). (b) Middle section of the SGP-based restored data (PSF parameters $\sigma_r = 106$ nm, $\sigma_z = 297$ nm). Insets show the magnified cross-sectional views of the volume denoted by the white box. The dotted lines indicate the position of the xy-, xz- and yz-slices shown for each of the 3D stacks. Scale bars: 10 μm and 1 μm (insets).

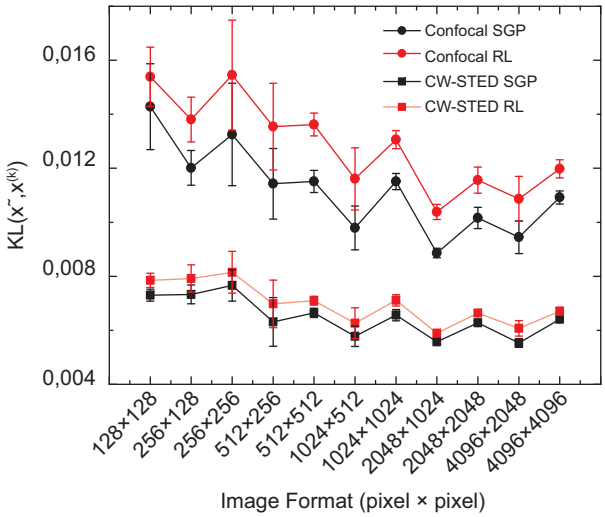


Figure 5.15: **Quality comparison between SGP and RL algorithms.** Kullback-Leibler divergence of the reconstructed image x_k from the known object $\boldsymbol{x}$ as a function of the image format. Same conditions of Figures 5.10 and 5.11.

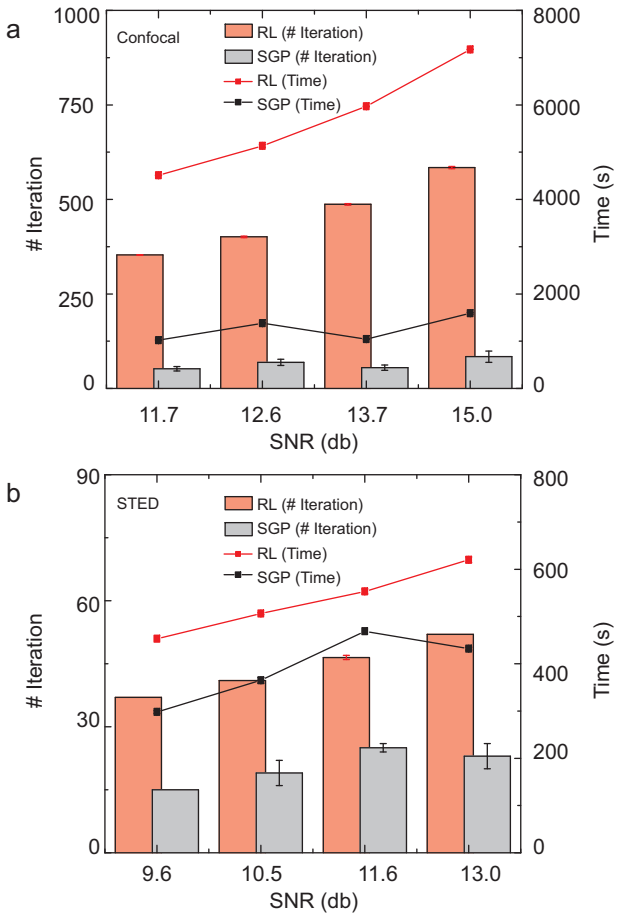


Figure 5.16: **Performance comparison between SGP and RL algorithms as function of SNR.** (a,b) Number of iteration and computational time as a function of the the SNR for CLSM (a) and STED microscopy (b). Same conditions of Figures 5.10 and 5.11. Image format 4096×4096. The SNR is controlled by changing the factor $\tau = 2, 3, 5, 10$, respectively.

5.5 ML estimation of regularization hyperparameters

In this section we show the numerical behaviour of the two phases gradient method proposed in Chapter 4 for the estimation of the regularization hyperparameters in inverse problems with wavelet regularization. The experiments are performed on synthetic test problems and a comparison with the Gradient Ascent method (Algorithm 4.1) is reported.

5.5.1 Gradient ascent and two phases gradient methods

In order to test our algorithm performances, as described in [28], we firstly randomly generate wavelet coefficients according to the *a priori* law (4.5) with realistic (for natural images) subband fixed parameters $\lambda_{\mathbf{m}}$. We use Symlets [40] of length 8 over $J = 2$ resolution levels and each subband $\mathbf{m}$ is represented by the triplet (j, l, c) where j is the resolution level index, and $(l, c)_{l\in\{0,1\},c\in\{0,1\}}$ represents the low/high-pass filtered subbands (the couple $(1, 1)$ thus represents the diagonal coefficients). The size of the generated image is 128×128, it is blurred using a Gaussian kernel A of standard deviation 0.5, and Gaussian noise is added (of variance $\sigma^2 = 25$).

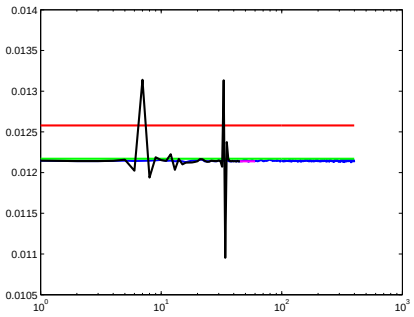
The classical gradient ascent algorithm is launched over 400 iterations using a fixed steplength $\alpha_n = 10^{-4}$ and the estimated parameters are computed as the mean value over the last 50 iterations. The two-phase algorithm uses these parameters: $P = 3, \theta = 0.5, \gamma = 10^{-4}$, $\alpha_{\max} = 10^7$, $\alpha_{\min} = 10^{-7}$, $\alpha_0 = 10^{-4}$; $N_1 = 50$, $N_2 = 10$ for the switching rule and $\tau_g = 10^{-2}, \tau_\lambda = 10^{-7}$ for the stopping rule. Both are initialized by applying a Wiener filter on g. In Figure 5.17 the behaviour over the iterations of the hyperparameters estimated by the two methods is shown. In Tab. 5.12 we report the number of iterations (it.) and gradient evaluations (Grad.), the relative error in Euclidean norm of the estimated parameters $\boldsymbol{\lambda}$ with respect to the theoretical values ($\mathrm{Err}_{th.}$) and the values estimated by the ML approach on the ground truth (Err_{ML}) and the time in seconds (time). All the test are executed in MATLAB on a quad core Intel i7 CPU.

We also considered two well-known images (Mandrill and Barbara) and generated their blurred images by Gaussian kernel A of standard deviation 2, and added Gaussian noise of variance $\sigma^2 = 25$. We applied the gradient methods to obtain estimated hyperparameters and used them to restore the image by means

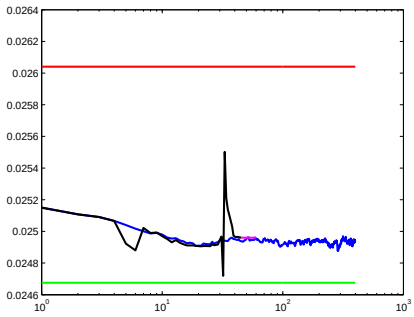
of a Forward-Backward algorithm [31]. In Tab. 5.13 we report the values of Signal to Noise Ratio (SNR) calculated on the corrupted image (SNR_{init}) and on the reconstruction obtained with the estimated $\boldsymbol{\lambda}$ (SNR_{fin}).

Figure 5.17: $\lambda_{\mathbf{m}}$ behavior over iterations. Green line is the ML value of $\lambda_{\mathbf{m}}$, red line is the theoretical value of $\lambda_{\mathbf{m}}$, blue line denotes the estimation provided by the GA method, black line and purple line refer to the estimations obtained in the first and second phase of the 2Ph algorithm.

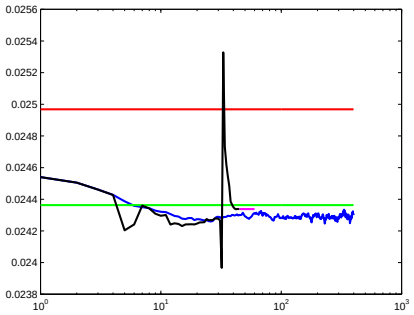
$\mathbf{m} = (2, 0, 0)$



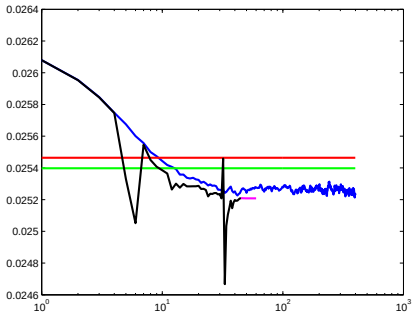
$\mathbf{m} = (2, 0, 1)$



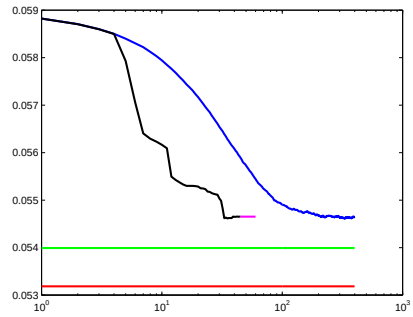
$\mathbf{m} = (2, 1, 0)$



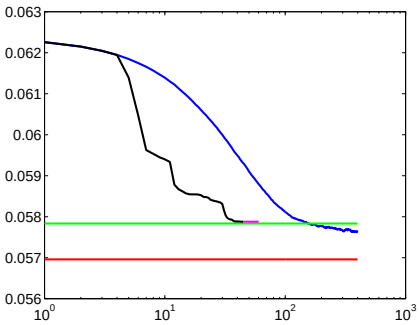
$\mathbf{m} = (2, 1, 1)$



$\mathbf{m} = (1, 0, 1)$



$\mathbf{m} = (1, 1, 0)$



$\mathbf{m} = (1, 1, 1)$

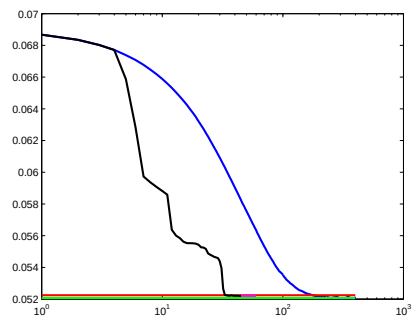


Table 5.12: Simulated Data

Alg.	It.	Grad.	Err_{ML}	$\text{Err}_{th.}$	time
2Ph	61	90	0.007	0.021	431.8
GA	400	400	0.007	0.020	1920.5

Table 5.13: Real Data

Problem	Alg.	It.	SNR_{init}	SNR_{fin}
Mandrill 256^2	2Ph	883	12.1616	14.2324
	GA	1000	12.1616	14.0051
Barbara 512^2	2Ph	785	18.5272	19.5316
	GA	1000	18.5272	19.3655

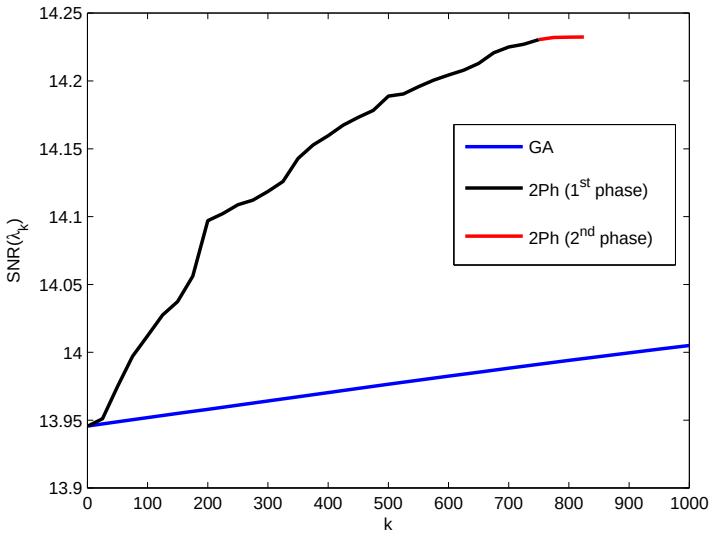


Figure 5.18: Evolution of SNR w.r.t. iterations for image Mandrill of size 256^2 with 2 levels of resolution

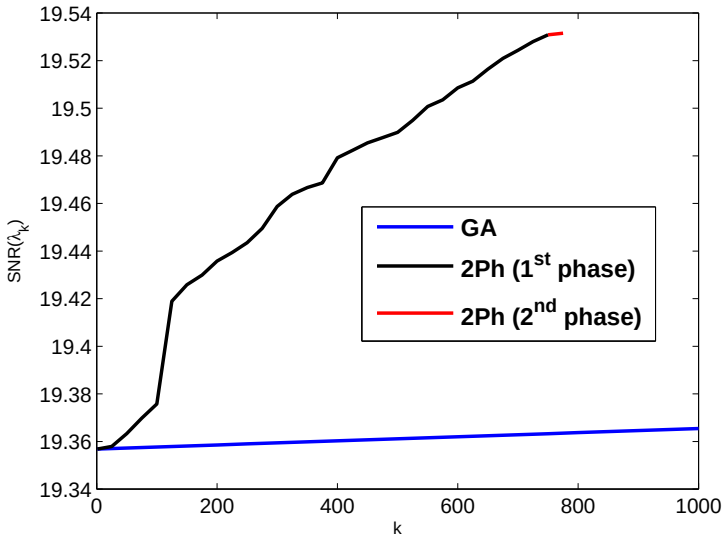


Figure 5.19: Evolution of SNR w.r.t. iterations for image Barbara of size 512^2 with 2 levels of resolution

Also, to verify the effectiveness of the method in obtaining a better reconstruction in fewer iterations than GA, we applied the reconstruction algorithm every 25 iterations of each algorithm and compared the SNR. The meaning of the different colours in Figures 5.18 and 5.19 are the following:

- black-red line: SNR correspondent to λ_k, taken every 25 iterations (change of colour means start of the second phase);
- blue line: SNR correspondent to λ_k, taken every 25 iterations, obtained by gradient ascent method with $\alpha_k = \alpha_0 = 1e-6$.

For both cases we used the algorithm 4.5 using the following parameter setting:

- nonmonotone line-search: $P = 4$;
- stopping tolerance: $\tau_g = 5 \cdot 10^{-1}$, $\tau_\lambda = 5 \cdot 10^{-7}$;
- maximum number of iteration: $maxit = 1000$;
- first phase: $N_1 = 750$, $N_2 = 30$ (black-red line);

5.5.2 Discussion

From Figure 5.17 we can observe remarkable acceleration of the hyperparameter estimation due to the suited choice of the steplength in the first phase and fruitful stabilization in the second phase given by the line-search strategy. Tables 5.12, 5.13 and Figures 5.18 and 5.19 confirm the promising convergence rate improvements provided by the 2Ph algorithm with respect to the standard gradient ascent method. These improvements are obtained without losing the accuracy of the proposed ML hyperparameter estimation and introducing the same steplength strategy exploited in the SGP method. However, we have to remark that each iteration is still computing-intensive and exhibits a quite high computational time. Nevertheless many recent papers [66, 121, 110] shows that accelerating the Monte Carlo Markov Chain (MCMC) method is possible and effective on GPUs, due to the component-wise operation needed in both generation of samples, evaluation of probability and computation of the acceptance ratio.

Conclusions

In this thesis the numerical solution of large-scale optimization problems arising in imaging applications is faced on different high performance computing architectures. In particular, optimization problems coming from image deblurring, image denoising and maximum likelihood approaches for estimating the regularization parameters are considered. Gradient-type methods are exploited for solving the optimization problems and parallel implementation for distributed memory multiprocessor systems and for Graphics Processing Units (GPU) are developed.

For solving the nonnegatively constrained optimization problems arising in image deblurring and denoising, scaled gradient projection methods are designed by using recent effective ideas for defining at each iteration the steplength parameter and the diagonal matrix used to scale the gradient direction. The updating of the steplength parameter is obtained by adaptive alternation of the Barzilai-Borwein rules while the scaling matrix is derived by a special splitting of the gradient of the objective function. The proposed Barzilai-Borwein steplength rule is successfully exploited also in designing accelerated gradient methods for the maximum likelihood estimation of the regularization hyperparameter in inverse problems with wavelet regularization. By means of these techniques, efficient optimization algorithms have been proposed that well compare with state of the art approaches on several imaging problems from astronomy and microscopy; however, in case of large-scale imaging problems (3D problems or multiple images problems), these algorithms are sometimes too much time-consuming and their parallel implementation suitable for fully exploiting modern multiprocessor architectures is welcome.

In this thesis we developed parallel versions of the scaled gradient projection methods able to combine the efficiency of the optimization solvers with the high performance hardware devices nowadays available on multiprocessors systems. In this way, we showed that real-time reconstructions of large images in both astron-

omy and microscopy can be achieved. In particular, popular non-expensive GPU devices have been fruitfully used for facing large-scale 3D deconvolution problems in microscopy and multiple imaging problems in astronomy. Finally, preliminary numerical experience suggests that also the challenging optimization problems arising in the regularization hyperparameter estimation could benefit from a suitable combination of the improvements in the numerical solvers and the enhancements due to the parallel implementation. The codes of the algorithms presented and discussed in this thesis for Matlab, IDL and IDL-CUDA languages are available at the URL http://www.unife/prin/software and can be freely downloaded, while the executables for the MPI and GPU versions can be requested contacting the authors.

Bibliography

[1] H.M. Adorf, R.N. Hook, L.B. Lucy, and F.D. Murtagh. Accelerating the richardson-lucy restoration algorithm. In *European Southern Observatory Conference and Workshop Proceedings*, volume 41, page 99, 1992.

[2] D. A. Agard. Optical sectioning microscopy: Cellular architecture in three dimensions. *Annual Review of Biophysics and Bioengineering*, 13:191–219, 1984.

[3] B. Anconelli, M. Bertero, P. Boccacci, C. Carbillet, and H. Lanteri. Restoration of interferometric images. efficient richardson-lucy methods for lincnirvana data reduction. *Astron. Astrophys.*, 430(2):731–738, 2005.

[4] B. Anconelli, M. Bertero, P. Boccacci, M. Carbillet, and H. Lanteri. Reduction of boundary effects in multiple image deconvolution with an application to LBT LINC–NIRVANA. *Astronomy & Astrophysics*, 448:1217–1224, 2006.

[5] B. Anconelli, M. Bertero, P. Boccacci, M. Carbillet, and H. Lanteri. Iterative methods for the reconstruction of astronomical images with hight dynamic range. *Journal of Computational and Applied Mathematics*, 198(2):321–331, 2007.

[6] C. Arcidiacono, E. Diolaiti, M. Tordi, R. Ragazzoni, J. Farinato, E. Vernet, and E. Marchetti. Layer-oriented simulation tool. *Applied optics*, 43(22):4288–4302, 2004.

[7] J. M. Bardsley and C.R. Vogel. A nonnegatively constrained convex programming method for image reconstruction. *SIAM Journal on Scientific Computing*, 25(4):1326–1343, 2003.

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

[9] J. Bect, L. Blanc-Féraud, G. Aubert, and A. Chambolle. A l^1-unified variational framework for image restoration. In T Pajdla and J Matas, editors,

Proceeding of Computer Vision-ECCV 2004, volume LNCS 3024, pages 1–13, Prague, Czech Republic, May 2004. Springer.

[10] F. Benvenuto, R. Zanella, L. Zanni, and M. Bertero. Nonnegative least-squares image deblurring: improved gradient projection approaches. *Inverse Problems*, 26(2):025004 (18pp), 2010.

[11] M. Bertero and P. Boccacci. Image restoration methods for the large binocular telescope (lbt). *Astronomy and Astrophysics Supplement Series*, 147:323–333, 2000.

[12] M. Bertero and P. Boccacci. A simple method for the reduction of boundary effects in the richardson-lucy approach to image deconvolution. *Astronomy & Astrophysics*, 437:369–374, 2005.

[13] M. Bertero and P. Boccacci. *Introduction to inverse problems in imaging*. CRC press, 2010.

[14] M. Bertero, P. Boccacci, G. J. Brakenhoff, F. Malfanti, and H. T. M. van der Voort. Three-dimensional image restoration and super-resolution in fluorescence confocal microscopy. *Journal of Microscopy*, 157:3–20, 1990.

[15] M. Bertero, P. Boccacci, G. Desiderà, and G. Vicidomini. Image deblurring with poisson data: from cells to galaxies. *Inverse Problems*, 25(12):123006, 2009.

[16] M. Bertero, P. Boccacci, A. La Camera, C. Olivieri, and M. Carbillet. Imaging with linc-nirvana, the fizeau interferometer of the large binocular telescope: state of the art and open problems. *Inverse Problems*, 27(11):113001, 2011.

[17] M. Bertero, P. Boccacci, G. Talenti, R. Zanella, and L. Zanni. A discrepancy principle for poisson data. *Inverse Problems*, 26(10):105004, 2010.

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

[19] D. S. C. Biggs and M. Andrews. Acceleration of iterative image restoration algorithms. *Applied optics*, 36(8):1766–1775, 1997.

[20] E. G. Birgin, J. M. Martinez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10(4):1196–1211, 2000.

[21] S. Bonettini. A nonmonotone inexact Newton method. *Optimization Methods and Software*, 20(4–5):475–491, 2005.

[22] S. Bonettini and M. Prato. Nonnegative image reconstruction from sparse fourier data: a new deconvolution algorithm. *Inverse Problems*, 26(9):095001, 2010.

[23] S. Bonettini, R. Zanella, and L. Zanni. A scaled gradient projection method for constrained image deblurring. *Inverse Problems*, 25(1):015002, 2009.

[24] M. A. Bruce and M. J. Butte. Real-time gpu-based 3d deconvolution. *Opt. Express*, 21(4):4766–4773, Feb 2013.

[25] M. Carbillet, S. Correia, P. Boccacci, and M. Bertero. Restoration of interferometric images. *A&A*, 387:744–757, 2002.

[26] M. Carbillet, S. Correia, P. Boccacci, and M. Bertero. Restoration of interferometric images. the case–study of the large binocular telescope. *Astronomy & Astrophysics*, 387(2):744–757, 2002.

[27] M. Carlavan and L. Blanc-Féraud. Sparse Poisson noisy image deblurring. *IEEE Trans. Image Processing*, 21(4):1834–1846, 2012.

[28] R. Cavicchioli, C. Chaux, L. Blanc-Féraud, and L. Zanni. Ml estimation of wavelet regularization hyperparameters in inverse problems. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1553–1557, May 2013.

[29] R. Cavicchioli, A. Prearo, R. Zanella, G. Zanghirati, and L. Zanni. Optimization methods for digital image restoration on mpp multicore architectures. *Advances in Nonlinear Optimization and Equilibrium Problems: a Tribute to Marco D'Apuzzo (V. De Simone, D. Di Serafino, G. Toraldo eds.) Quaderni di Matematica, Dip. di Matematica della Seconda Universit di Napoli, Aracne*, 27:93–116, 2012.

[30] C. Chaux and L. Blanc-Féraud. Wavelet-based hyperparameter estimation for solving inverse problems. Technical report, 2012.

[31] C. Chaux, P. L. Combettes, J.-C. Pesquet, and V. R. Wajs. A variational formulation for frame based inverse problems. *Inverse Problems*, 23:1495–1518, June 2007.

[32] W. Cheng and Z. Chen. Nonmonotone spectral method for large-scale symmetric nonlinear equations. *Numerical Algorithms*, 62(1):149–162, 2013.

[33] P. L. Combettes and J.-C. Pesquet. A Douglas-Rachford Splitting Approach to Nonsmooth Convex Variational Signal Recovery. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):564–574, 2007.

[34] J. A. Conchello and J. G. McNally. Fast regularization technique for expectation maximization algorithm for optical sectioning microscopy. In Carol J. Cogswell Gordon S. Kino Tony Wilson, editor, *Three-Dimensional Microscopy: Image Acquisition and Processing III*, volume 2655, pages 199–208. SPIE, 1996.

[35] Y. H. Dai. Alternate step gradient method. *Optimization*, 52(4–5):395–415, 2003.

[36] Y. H. Dai and R. Fletcher. New Algorithms for Singly Linearly Constrained Quadratic Programming Problems Subject to Lower and Upper Bounds. *Mathematical Programming*, 106(3):403–421, 2006.

[37] Y. H. Dai, W. W. Hager, K. Schittkowski, and H. Zhang. The cyclic Barzilai-Borwein method for unconstrained optimization. *IMA J. Numer. Anal.*, 26:604–627, 2006.

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

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

[40] I. Daubechies. *Ten Lectures on Wavelets*. CBMS-NSF, SIAM Lecture Series, Philadelphia, PA, USA, 1992.

[41] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Applied Math.*, 57:1413–1457, 2004.

[42] C. A. Deledalle, S. Vaiter, G. Peyré, J. Fadili, and C. Dossal. Unbiased Risk Estimation for Sparse Analysis Regularization. In *Proc. Int. Conf. Image Process.*, Orlando, Florida, USA, 2012.

[43] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39 (series B):1–38, 1977.

[44] N. Dey, L. Blanc-Féraud, C. Zimmer, P. Roux, Z. Kam, J. C. Olivo-Marin, and J. Zerubia. Richardson-lucy algorithm with total variation regularization for 3d confocal microscope deconvolution. *Microsc. Res. Tech.*, 69:260–266, 2006.

[45] A Diaspro, editor. *Confocal and Two-Photon Microscopy: Foundations, Applications, and Advances*. John Wiley & Sons, 2002.

[46] A. Diaspro, P. Bianchini, G. Vicidomini, M. Faretta, P. Ramoino, and C. Usai. Multi-photon excitation microscopy. *BioMedical Engineering OnLine*, 5(1):36, 2006.

[47] F. Difato, F. Mazzone, S. Scaglione, M. Fato, F. Beltrame, L. Kubnov, J. Jancek, P. Ramoino, G. Vicidomini, and A. Diaspro. Improvement in volume estimation from confocal sections after image deconvolution. *Microsc. Res. Tech.*, 64(2):151–155, 2004.

[48] G. Donnert, J. Keller, C. Wurm, S. Rizzoli, V. Westphal, A. Schonle, R. Jahn, S. Jakobs, C. Eggeling, and S. Hell. Two-Color Far-Field Fluorescence Nanoscopy. *Biophys. J.*, 92(8):L67—-L69, 2007.

[49] F.-X. Dupé, J. Fadili, and J.-L. Stark. A proximal iteration for deconvolving poisson noisy images using sparse representations. *IEEE Trans. Image Process.*, 18:310–321, 2009.

[50] P. Favati, G. Lotti, O. Menchi, and F. Romani. Performance analysis of maximum likelihood methods for regularization problems with nonnegativity constraints. *Inverse Problems*, 26:085013 (18pp), 2010.

[51] M. A. T. Figueiredo and R. D. Nowak. An EM Algorithm for Wavelet-Based Image Restoration. *IEEE Trans. on Image Proc.*, 12(8):906–916, 2003.

[52] R. Fletcher. On the Barzilai-Borwein Method. Technical Report NA/207, University of Dundee, 2001.

[53] G. Frassoldati, G. Zanghirati, and L. Zanni. New Adaptive Stepsize Selections in Gradient Methods. *J. Indutrial and Management Optimization*, 4(2):299–312, 2008.

[54] A. Friedlander, J. M. Martinez, B. Molina, and M. Raydan. Gradient method with retards and generalizations. *SIAM Journal on Numerical Analysis*, 36(1):275–289, 1998.

[55] S. Frisken Gibson and F. Lanni. Experimental test of an analytical model of aberration in an oil-immersion objective lens used in three-dimensional light microscopy. *J. Opt. Soc. Am. A*, 9(1):154–166, 1992.

[56] N. P. Galatsanos and A. K. Katsaggelos. Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation. *IEEE Trans. on Image Proc.*, 1(3):322–336, 1992.

[57] D. Geman and C. Yang. Nonlinear Image recovery with Half-Quadratic Regularization. *IEEE Transactions on Image Processing*, 23(7):932–946, 1995.

[58] M. D. Gonzalez-Lima, W. W. Hager, and H. Zhang. An affine-scaling interior-point method for continuous knapsack constraints. Technical Report CCT-TR-2009-10, Louisiana State University, Center for Computation {&} Technology, 2009.

[59] Joseph W. Goodman. *Introduction to Fourier optics*. Roberts and Company Publishers, 2005.

[60] J. P. Green. On use of the EM algorithm for penalized likelihood estimation. *J. Roy. Stat. Soc. B.*, 52(3):443–452, 1990.

[61] L. Grippo, F. Lampariello, and S. Lucidi. A nonmonotone line–search technique for Newton's method. *SIAM Journal on Numerical Analysis*, 23(4):707–716, 1986.

[62] G.-Z. Gu, D.-H. Li, L. Qi, and S.-Z. Zhou. Descent directions of quasi-newton methods for symmetric nonlinear equations. *SIAM Journal on Numerical Analysis*, 40(5):1763–1774, 2002.

[63] M. G. L. Gustafsson. Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy. *Journal of Microscopy*, 198(2):82–87, 2000.

[64] S. B. Hadj, L. Blanc-Féraud, G. Aubert, and G. Engler. Blind restoration of confocal microscopy images in presence of a depth-variant blur and poisson noise. In *Proc. ICASSP 2013*, pages 915–919. IEEE, 2013.

[65] W. W. Hager, B. A. Mair, and H. Zhang. An affine-scaling interior-point CBB method for box-constrained optimization. *Math. Program.*, 119(1):1–32, 2009.

[66] C. Hall, W. Ji, and Estela. Blaisten-Barojas. The metropolis monte carlo method with cuda enabled graphic processing units. *Journal of Computational Physics*, 258:871–879, 2014.

[67] B. Harke, J. Keller, C. K. Ullal, V. Westphal, A. Schönle, and S. W. Hell. Resolution scaling in STED microscopy. *Opt. Express*, 16(6):4154–4162, 2008.

[68] R. Heintzmann and M. G. L. Gustafsson. Subdiffraction resolution in continuous samples. *Nature Photon.*, 3(7):362–364, 2009.

[69] S. W. Hell. Microscopy and its focal switch. *Nat. Methods*, 6(1):24–32, 2009.

[70] S. W. Hell, E. H. K. Stelzer, S. Lindek, and C. Cremer. Confocal microscopy with an increased detection aperture: type-B 4Pi confocal microscopy. *Opt. Lett.*, 19(3):222–224, 1994.

[71] T. Herbst, R. Ragazzoni, D. Andersen, H. Boehnhardt, P. Bizenberger, A. Eckart, W. Gaessler, H.-W. Rix, R.-R. Rohloff, P. Salinari, et al. Lincnirvana: a fizeau beam combiner for the large binocular telescope. In *Proceedings of SPIE*, volume 4838, pages 456–465, 2003.

[72] T. J. Holmes. Blind deconvolution of quantum-limited incoherent imagery: maximum-likelihood approach. *J. Opt. Soc. Am. A*, 9(7):1052–1061, 1992.

[73] B. Huang, H. Babcock, and X. Zhuang. Breaking the Diffraction Barrier: Super-Resolution Imaging of Cells. *Cell*, 143(7):1047–1058, 2010.

[74] H. M. Hudson and R. S. Larkin. Accelerated image reconstruction using ordered subsets of projection data. *Medical Imaging, IEEE Transactions on*, 13(4):601–609, 1994.

[75] J. Huisken and D. Y. R. Stainier. Selective plane illumination microscopy techniques in developmental biology. *Development*, 136(12):1963–1975, 2009.

[76] J. Huisken, J. Swoger, F. Del Bene, J. Wittbrodt, and E. H. K. Stelzer. Optical Sectioning Deep Inside Live Embryos by Selective Plane Illumination Microscopy. *Science*, 305(5686):1007–1009, 2004.

[77] M. M. Ichir and A. Mohammad-Djafari. Hidden Markov models for wavelet-based blind source separation. *IEEE Trans. on Image Proc.*, 15(7):1887–1899, 2006.

[78] A. N. Iusem. Convergence analysis for a multiplicatively relaxed {EM} algorithm. *Math. Methods Appl. Sci.*, 14(8):573–593, 1991.

[79] A. Jalobeanu, L. Blanc-Féraud, and J. Zerubia. Hyperparameter estimation for satellite image restoration using a MCMC Maximum Likelihood method. *Pattern Recognition*, 35(2):341–352, 2002.

[80] M. Jansen and A. Bultheel. Multiple wavelet threshold estimation by generalized cross validation for images with correlated noise. *IEEE Trans. on Image Proc.*, 8(7):947–953, July 1999.

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

[82] W. La Cruz, J. M. Martínez, and M. Raydan. Spectral Residual Method Without Gradient Information for Solving Large-Scale Nonlinear Systems of Equations. *Mathematics of Computation*, 75(255):1429, 2006.

[83] G. Landi and E. Loli Piccolomini. A projected newton-cg method for nonnegative astronomical image deblurring. *Numerical Algebra*, 48(4):279–300, 2008.

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

[85] H. Lantèri, M. Roche, O. Cuevas, and C. Aime. A general method to devise maximum-likelihood signal restoration multiplicative algorithms with nonnegativity constraints. *Signal Process.*, 81:945–974, 2001.

[86] T. Le, R. Chartran, and T. Asaki. A variational approach to reconstructing images corrupted by poisson noise. *J. Math. Imaging Vision*, 27:257–263, 2007.

[87] S. Lee and S. J. Wright. Implementing algorithms for signal and image reconstruction on graphical processing units. *Computer Sciences Department, University of Wisconsin-Madison, Tech. Rep*, 2008.

[88] D. Leporini and J.-C. Pesquet. Bayesian wavelet denoising: Besov priors and non-Gaussian noises. *Signal processing*, 81(1):55–67, 2001.

[89] J. Llacer and J. Nunez. Iterative maximum likelihood estimator and bayesian algorithms for image reconstruction in astronomy. In R. L. White and R. J. Allen, editors, *Restoration of HST Images and Spectra*, pages 52–70. Space Telescope Science Institute, Baltimore, MD, 1990.

[90] L. B. Lucy. An iterative technique for the rectification of observed distributions. *Astronom. J.*, 79:745–754, 1974.

[91] L.B. Lucy and R.N. Hook. Co-adding images with different psf's. In *Astronomical Data Analysis Software and Systems I*, volume 25, page 277, 1992.

[92] S. Mallat. *A wavelet tour of signal processing*. Academic Press, San Diego, USA, 1998.

[93] P.P. Mondal, G. Vicidomini, and A. Diaspro. Markov random field aided bayesian approach for image reconstruction in confocal microscopy. *Journal of Applied Physics*, 102(4), 2007.

[94] P.P. Mondal, G. Vicidomini, and A. Diaspro. Image reconstruction for multiphoton fluorescence microscopy. *Applied Physics Letters*, 92(10), 2008.

[95] E. A. Mukamel, H. Babcock, and X. Zhuang. Statistical deconvolution for superresolution fluorescence microscopy. *Biophys. J.*, 102(10):2391–2400, May 2012.

[96] M. A. A. Neil, R. Juskaitis, and T. Wilson. Method of obtaining optical sectioning by using structured light in a conventional microscope. *Opt. Lett.*, 22(24):1905–1907, December 1997.

[97] NVidia. *NVIDIA CUDA Compute Unified Device Architecture, Programming Guide*, version 2. edition, 2008.

[98] J. B. Pawley, editor. *Handbook of Biological Confocal Microscopy*. Springer, 3rd edition, 2006.

[99] M. Prato, R. Cavicchioli, L. Zanni, P. Boccacci, and M. Bertero. Efficient deconvolution methods for astronomical imaging: algorithms and idl-gpu codes. *Astronomy & Astrophysics*, 539:133, 2012.

[100] C. Preza and J.-A. Conchello. Depth-variant maximum-likelihood restoration for three-dimensional fluorescence microscopy. *J. Opt. Soc. America A*, 21:1593–1601, 2004.

[101] W. H. Richardson. Bayesian–based iterative method of image restoration. *J. Opt. Soc. Amer. A*, 62(1):55–59, 1972.

[102] C. Robert. *Discretization and MCMC convergence assessment*. Lecture Notes in Statistics. Springer-Verlag New York Inc., New York, 1998.

[103] V. Ruggiero, T. Serafini, R. Zanella, and L. Zanni. Iterative regularization algorithms for constrained image deblurring on graphics processors. *Journal Of Global Optimization*, pages 1–13, 2010.

[104] T. Serafini, R. Zanella, and L. Zanni. Gradient projection methods for image deblurring and denoising on graphics processors. *Int. Conf. on Parallel Computing "ParCo2009", Advances in Parallel Computing*, 19:95–66, 2010.

[105] T. Serafini, G. Zanghirati, and L. Zanni. Gradient projection methods for quadratic programs and applications in training support vector machines. *Optimization Methods and Software*, 20(2–3):343–378, 2005.

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

[107] J.-B. Sibarita. Deconvolution microscopy. In Theodorus W. J. Gadella Jens Rietdorf, editor, *Microscopy Techniques*, pages 201–243. Springer-Verlag, 2005.

[108] 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.

[109] 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.

[110] D. N. VanDerwerken and S. C. Schmidler. Parallel markov chain monte carlo. *arXiv preprint arXiv:1312.7479*, 2013.

[111] P. J. Verveer, J. Swoger, F. Pampaloni, K. Greger, M. Marcello, and E. H. K. Stelzer. High-resolution three-dimensional imaging of large specimens with light sheet-based microscopy. *Nat. Methods*, 4(4):311–313, 2007.

[112] G. Vicidomini, P. Boccacci, A. Diaspro, and M. Bertero. Application of the split-gradient method to 3D image deconvolution in fluorescence microscopy. *Journal of Microscopy*, 234:47–61, 2009.

[113] G. Vicidomini, M. C. Gagliani, K. Cortese, J. Krieger, P. Buescher, P. Bianchini, P. Boccacci, C. Tacchetti, and A. Diaspro. A novel approach for correlative light electron microscopy analysis. *Microscopy Research and Technique*, 73(3):215–224, 2010.

[114] G. Vicidomini, I. C. Hernández, M. dAmora, F. Cella Zanacchi, P. Bianchini, and A. Diaspro. Gated cw-sted microscopy: A versatile tool for biological nanometer scale investigation. *Methods*, (0), 2013.

[115] G. Vicidomini, G. Moneron, C. Eggeling, E. Rittweger, and S. W. Hell. STED with wavelengths closer to the emission maximum. *Optics Express*, 20(5):5225–5236, February 2012.

[116] G. Vicidomini, G. Moneron, K.Y. Han, V. Westphal, H. Ta, M. Reuss, J. Engelhardt, and S.W. Eggeling, C.and Hell. Sharper low-power sted nanoscopy by time gating. *Nature Methods*, 8(7):571–575, 2011. cited By (since 1996) 11.

[117] G. Vicidomini, R. Schmidt, A. Egner, S. Hell, and A. Schönle. Automatic deconvolution in 4pi-microscopy with variable phase. *Optics Express*, 18(10):10154–10167, May 2010.

[118] G. Vicidomini, A. Schönle, H. Ta, K. Y. Han, G. Moneron, C. Eggeling, and S. W. Hell. Sted nanoscopy with time-gated detection: Theoretical and experimental aspects. *PLoS ONE*, 8(1):e54421, 01 2013.

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

[120] C. Vonesch, S. Ramani, and M. Unser. Recursive risk estimation for nonlinear image deconvolution with a wavelet-domain sparsity constrain. In

Proc. Int. Conf. Image Process., pages 665–668, San Diego CA, USA, Oct. 12-15 2008.

[121] G. White and M. D. Porter. Gpu accelerated mcmc for modeling terrorist activity. *Computational Statistics & Data Analysis*, 71(C):643–651, 2014.

[122] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Trans. Signal Process.*, 57(7):2479–2493, 2009.

[123] L. Younes. Parametric inference for imperfectly observed Gibbsian fields. *Probab. Theory Relat. Fields*, 82:625–645, 1989.

[124] R. Zanella, P. Boccacci, L. Zanni, and M. Bertero. Efficient gradient projection methods for edge-preserving removal of Poisson noise. *Inverse Problems*, 25(4):45010, 2009.

[125] R. Zanella, G. Zanghirati, R. Cavicchioli, L. Zanni, P. Boccacci, M. Bertero, and G. Vicidomini. Towards real-time image deconvolution: application to confocal and sted microscopy. *Scientific reports*, 3, 2013.

[126] L. Zanni. An improved gradient projection-based decomposition technique for support vector machines. *Computational Management Science*, 3:131–145, 2006.

[127] B. Zhang, J. Zerubia, and J.-C. Olivo-Marin. Gaussian approximations of fluorescence microscope point-spread function models. *Appl. Opt.*, 46(10):1819–1829, April 2007.

[128] B. Zhou, L. Gao, and Y. H. Dai. Gradient methods with adaptive step-sizes. *Comput. Optim. Appl.*, 35(1):69–86, 2006.

[129] B. Zhou, L. Gao, and Y. H. Dai. Monotone projected gradient methods for large-scale box-constrained quadratic programming. *Science in China: Series A Mathematics*, 49(5):688–702, 2006.