UNIVERSITY OF INSUBRIA

Department of Science and High Technology

Ph.D. course in Computer Science and Computational Mathematics (XXIX cycle)

Tikhonov-type iterative regularization methods for ill-posed inverse problems: theoretical aspects and applications

Author:
Alessandro BUCCINI

Supervisor:
Prof. Marco DONATELLI

"I accept nothing on authority. A hypothesis must be backed by reason, or else it is worthless."

Isaac Asimov

UNIVERSITY OF INSUBRIA

Abstract

Department of Science and High Technology

Doctor of Philosophy

Tikhonov-type iterative regularization methods for ill-posed inverse problems: theoretical aspects and applications

by Alessandro BUCCINI

Ill-posed inverse problems arise in many fields of science and engineering. The ill-conditioning and the big dimension make the task of numerically solving this kind of problems very challenging.

In this thesis we construct several algorithms for solving ill-posed inverse problems. Starting from the classical Tikhonov regularization method we develop iterative methods that enhance the performances of the originating method.

In order to ensure the accuracy of the constructed algorithms we insert a priori knowledge on the exact solution and empower the regularization term, thus keeping under control the ill-conditioning of the problems. By exploiting the structure of the problem we are also able to achieve fast computation even when the size of the problem becomes very big.

The methods we developed, in order to be usable for real-world data, need to be as free of parameters as possible. For most of the proposed algorithms we provide efficient strategies for the choice of the regularization parameters, which, most of the times, rely on the knowledge of the norm of the noise that corrupts the data.

We construct algorithms that enforce constraint on the reconstruction, like nonnegativity or flux conservation and exploit enhanced version of the Euclidian norm using a regularization operator and different semi-norms, like the Total Variaton, for the regularization term.

For each method we analyze the theoretical properties, like, convergence, stability, and regularization. Depending on the method we are going to consider the finite dimensional case or the more general case of Hilbert spaces.

Numerical examples prove the good performances of the algorithms proposed in term of both accuracy and efficiency. We consider different kinds of mono-dimensional and two-dimensional problems, with a particular attention to the restoration of blurred and noisy images.

Acknowledgements

First and foremost I would like to thank my supervisor Prof. Marco Donatelli for his continuous support during these three years of Ph.D.. He taught me a lot; not only he has allowed me to grow as a mathematician, but also as a person.

Thanks to all the people I had the privilege to work with: Zhong-Zhi Bai, Davide Bianchi, Pietro Dell'Acqua, Fabio Ferri, Ken Hayami, Ronny Ramlau, Lothar Reichel, Stefano Serra-Capizzano, Jun-Feng Yin, and, Ning Zheng.

I would also like to thank the referees Marco Prato and Lothar Reichel for their remarks, which have improved the quality and the readability of this Ph.D. thesis.

A special thanks to Lothar and Laura for their hospitality and kindness during my staying in Kent. You really made me feel at home!

Thanks also to Ronny for having me in Linz. It had been a wonderful time!

I would also wish to thank all the *Insubria family* who has always been there when I needed them most.

Finally, I would like to thank my family for always supporting and understanding me.

Contents

List of Figures

List of Tables

List of Abbreviations

ADMM	**A**lternating **D**irection **M**ultiplier **M**ethod
AIT	**A**pproximated **I**terated **T**ikhonov
AIT-GP	**A**pproximated **I**terated **T**ikhonov with **G**eneral **P**enalty term
APIT	**A**pproximated **P**rojected **I**terated **T**ikhonov
APIT-GP	**A**pproximated **P**rojected **I**terated **T**ikhonov with **G**eneral **P**enalty term
CGLS	**C**onjugate **G**radient method for **L**east **S**quare problems
CSeB-A	**C**omputational **Se**mi**B**lind **A**DMM
FlexiAT	**Flexi**ble **A**rnoldi **T**ikhonov
FOV	**F**ield **O**f **V**iew
GIT	**I**terated **T**ikhonov with **G**eneral penalty term
GIT$_{NS}$	**N**on**s**tationary **I**terated **T**ikhonov with **G**eneral penalty term
GSVD	**G**eneralized **S**ingular **V**alue **D**ecomposition
IT	**I**terated **T**ikhonov
IT$_{NS}$	**N**on**s**tationary **I**terated **T**ikhonov
LCP	**L**inear **C**omplementarity **P**roblem
LR	**L**ucy-**R**ichardson
LSQR	**L**east **Sq**uares **R**esidual method
MvPs	**M**atrix-**v**ector **P**roduct**s**
MgM	**M**ulti**g**rid **M**ethod
MM	**M**odulus **M**ethod
NN-Restart-GAT	**N**on**n**egative **Restart**ed **G**eneralized **A**rnoldi **T**ikhonov
NSIFT	**N**on**s**tationary **I**terated **F**ractional **T**ikhonov
NSIWT	**N**on**s**tationary **I**terated **W**eighted **T**ikhonov
NNLS	**N**on**n**egative constrained **L**east **S**quares
NNQP	**N**on**n**egative constrained **Q**uadratic **P**rogramming
PDE	**P**artial **D**ifferential **E**quation
PSF	**P**oint **S**pread **F**unction
PSNR	**P**eak **S**ignal to **N**oise **R**atio
RRAT	**R**ange **R**estricted **A**rnoldi **T**ikhonov
RRE	**R**elative **R**econstruction **E**rror
SeB-A	**Se**mi**B**lind **A**DMM
SIFT	**S**tationary **I**terated **F**ractional **T**ikhonov
SIWT	**S**tationary **I**terated **W**eighted **T**ikhonov
SNR	**S**ignal to **N**oise **R**atio
SVD	**S**ingular **V**alue **D**ecomposition
TwIST	**Tw**o **S**teps **I**terative **S**hrinkage **T**hresholding
WCLR	**W**eakly **C**onstrained **L**ucy-**R**ichardson
wlsc	**W**eakly **L**ower **S**emi-**c**ontinuous

List of Symbols

$\|\mathbf{x}\|_1$	1 norm of $\mathbf{x}$
A^*	Adjoint operator (for matrices it denotes the conjugate transpose of A)
$\operatorname{argmin}_{x\in\Omega} f(x)$	Argument of the minimum of the function $f(x)$ on the set Ω[1]
$\mathcal{L}_A$	Augmented Lagrangian
$\mathbf{x} * \mathbf{y}$	Convolution between $\mathbf{x}$ and $\mathbf{y}$
$\mathcal{D}(A)$	Domain of A
$A \circ B$	Element-wise multiplication between matrices
$\|\mathbf{x}\|$	Euclidean norm of $\mathbf{x}$
$\nabla_s \phi(s,t)$	Gradient of ϕ with respect to the variable s
H_1	H_1 Sobolev space
$\mathcal{X}, \mathcal{Y}$	Hilbert spaces
$\inf_{x\in\Omega} f(x)$	Infimum of the function $f(x)$ on the set Ω[1]
$\langle \mathbf{x}, \mathbf{y} \rangle$	Inner product between $\mathbf{x}$ and $\mathbf{y}$
$\max_{x\in\Omega} f(x)$	Maximum of the function $f(x)$ on the set Ω[1]
$\min_{x\in\Omega} f(x)$	Minimum of the function $f(x)$ on the set Ω[1]
$\sup_{x\in\Omega} f(x)$	Supremum of the function $f(x)$ on the set Ω[1]
$\|\mathbf{x}\|_{TV}$	Total Variation norm of $\mathbf{x}$
A^t	Transpose of A
P_Ω	Metric projection over Ω
$A^\dagger$	Moore-Penrose Pseudoinverse of A
$\mathcal{N}(A)$	Null space of A
$(x)_+$	Positive part of x, i.e., $(x)_+ = \max\{x, 0\}$
$\mathcal{R}(A)$	Range of A
$\operatorname{Re}(x)$	Real part of x
$A\big\|_\Omega$	Restriction of A on Ω
$\sigma(A)$	Set of the singular values of A
$\lambda(A)$	Spectrum of A
$\partial_s \phi(s,t)$	Subdifferential of ϕ with respect to the variable s

[1] If Ω is not specified then Ω is the whole domain of f

To my parents, my family, and friends.
To Dr. V. Colombo who saved my world three times.

Chapter 1

Introduction

In this thesis we deal with *ill-posed inverse problems*. This kind of problems arises in many scientific fields from mathematics to physics and engineering. Solving these problems can be very difficult, but they present interesting challenges from both the theoretical and computational point of view; see [61] for more details about inverse problems.

There is no formal definition of inverse problems. Intuitively we are dealing with an inverse problem when we want to recover an object from some measured data knowing the process that generated the latter from the first.

We mainly consider problems that, when discretized, lead to linear system of equations. Because the original problem is ill-posed the resulting linear system is severely ill-conditioned. In real applications, it is impossible to avoid the presence of noise in the data, i.e., the right-hand side of the system, so direct inversion leads to very poor reconstructions. These problems usually have very large dimensions making the work at hand even more complicated. Moreover, to enhance the quality of the computed solution, we enforce constraints, like non-negativity, furtherly complicating the numerical methods involved in the solution of the problem.

Consider the case in which the operator to be inverted is compact. It is well known that the inverse of such an operator is unbounded. In particular, when the data is noise affected, the inversion process will amplify the noise to the point of corrupting the entire reconstruction making it completely useless.

In order to recover useful solutions, we need to resort to regularization methods, see e.g. [61, 78, 80]. Regularization methods substitute the original ill-conditioned problem with a well-conditioned one whose solution is a good approximation of the original. The accuracy of the recovered solution depends, at least in part, on how much information is available on the true solution and how this knowledge is inserted inside the algorithm itself.

The goal of this thesis is to develop several regularization methods for solving ill-posed inverse problems. For each of the algorithm proposed we give a theoretical analysis of their properties and show the performances on synthetic data.

The keystone of the methods we are going to develop is Tikhonov regularization which is described in Section 2.2. We are going to use the basic idea of Tikhonov regularization in order to formulate new and accurate iterative methods. The regularization effect will be obtained either by the knowledge of the limit point of such iterations or by the early stop of these by means of a suitable stopping criterion.

We are also going to consider strategies for achieving fast computations, either by exploiting the structure of the problem or by using linear algebra techniques, like Krylov methods, to compress the dimension of the problem without losing any relevant information.

The formulation of these new methods will be obtained by using the available knowledge on the exact solution and by improving the effectiveness of the regularization terms.

For testing the quality of our methods we will consider both one and two dimensional problems. In particular, for most of this thesis, we are going to consider the case of image deblurring.

The task of recovering an image from a blurred version of itself is a classical application in the framework of inverse problems. The blurring phenomenon can be modeled as a Fredholm integral of the first kind where the kernel is compact and possibly smooth. This operator reduces itself to a convolution operator when the blur is assumed to be spatially invariant, i.e., when the blur does not depend on the location. In the spatially invariant case the discretized operator can be represented by a highly structured matrix. In real case scenarios it is possible to have knowledge only over a finite region, the so called Field of View (FOV). In order to avoid underdetermined linear system it is necessary to make assumption on what is outside the FOV by means of the boundary conditions, see [84] for more details. The structure of the blurring operator is determined by the boundary conditions, but the basic structure is that of two level Toeplitz, i.e, a block Toeplitz with Toeplitz block matrix, where we recall that a Toeplitz matrix is a matrix whose entries are constant along the diagonals. This structure gives us the possibility ti exploit the Fast Fourier transform (FFT) to lower the computational effort of the algorithms. Moreover, Toeplitz matrices have been widely studied and thus we have a very deep knowledge of their properties.

In many situations it is known that the exact solution of the problem lies in some set. It might then be helpful to constrain the reconstructions to lie inside this set. In the case of image deblurring, for example, it is well known that the solution cannot attain negative value, so constraining the reconstruction to belong to the nonnegative cone can greatly improve the quality of the reconstruction. We are going to see that inserting this kind of knowledge inside the algorithms can enhance the quality of the reconstruction while having a very small impact on the computational cost.

This thesis is structured as follows

Chapter 2. Background In this chapter we give an insight on the basic concept of ill-posed problems. We formally introduce the image deblurring problem and explore some of his aspects, like the boundary conditions. We then describe the Tikhonov regularization method in both its standard and general form, where the regularization term is measured with a semi-norm usually defined by the discretization of a differential operator. Finally, we derive the iterated Tikhonov (IT) method as a refinement technique solving the error equation by Tikhonov regularization.

Chapter 3. Constrained Tikhonov Minimization As stated above, the introduction of a constraint inside the regularization can improve the quality of the reconstruction. In this chapter we analyze the case of nonnegatively constrained Tikhonov regularization. We reformulate the problem at hand in a suitable way in order to be able to apply the Modulus Method (MM). This method let us compute the solution of the constrained problem. In order to reduce the computational effort, we use the Golub-Kahan bidiagonalization technique to project the problem into a Krylov subspace of fairly small dimension. In this way we are able to obtain a very fast method without losing anything in term of quality of the reconstruction. The contents of this chapter are based on [8].

Chapter 4. Iterated Tikhonov with general penalty term The theory of the IT method has been developed only in the case where Tikhonov in its standard form is considered. In this chapter we develop a theory for the IT algorithm where Tikhonov is considered in its general form. We consider both the stationary and nonstationary version of the algorithm. We analyze its convergence in the noise-free case and show that in the noisy case, if equipped with a suitable stopping criterion, this method is a regularization method. Comparison with the classical IT algorithm shows how much the usage of the general form helps in improving the quality of the provided reconstruction. This chapter is related to the paper [30].

Chapter 5. Fractional and Weighted Iterated Tikhonov In two recent works [86, 95] two extensions of the classical Tikhonov regularization method were introduced. We provide saturation and converse results on their convergence rate. We formulate, using a refinement technique, the related iterative method both in stationary and nonstationary version. We show that these iterated methods are of optimal order and overcome the previous saturation results. Furthermore, for nonstationary iterated fractional Tikhonov regularization methods, we establish their convergence rate under general conditions on the iteration parameters. Numerical results confirm the improvements obtained against the classical IT algorithm. The contents of this chapter are based on [14].

Chapter 6. Approximated Iterated Tikhonov: some extensions The nonstationary preconditioned iteration proposed in the recent work [49] can be seen as an approximated iterated Tikhonov method. Starting from this observation in this chapter we extend the previous iteration in two directions: the usage of Tikhonov in its general form, as suggested by Chapter 4, and the projection into a convex set (e.g., the nonnegative cone as suggested by the results in Chapter 3). Depending on the application both generalizations can lead to an improvement in the quality of the computed approximations. Convergence results and regularization properties of the proposed iterations are proved. Finally, the new methods are applied to image deblurring problems and compared with the iteration in the original work and other methods with similar properties recently proposed in the literature. The contents of this chapter are taken from [26].

Chapter 7. Multigrid iterative regularization method Multigrid methods have been successfully used for solving linear systems coming from the discretization of PDEs for many years. It has been only in recent years that they have been considered for ill-posed problems. Their regularization power has been discussed for the first time in [56]. In this chapter we construct a multigrid algorithm for image deblurring that combines both linear and non-linear methods. The grid transfer operator used for this method is able to preserve the structure of the operator across the levels making possible to achieve fast computations. We combine one of the algorithms described in Chapter 6 with Linear Framelet Denoising to regularize the problem while preserving the details of the image without amplifying the noise. We study the convergence of the algorithm under some restrictive, but reasonable, hypothesis and prove that, if provided with the suitable stopping criterion, it is a regularization method. The comparison with other methods from the literature proves that it is a very powerful method able to restore with high accuracy blurred image without having to estimate any parameter. The contents of this chapter are taken from [27].

Chapter 8. Weakly Constrained Lucy-Richardson Lucy-Richardson (LR) is a classical iterative regularization method largely used for the restoration of nonnegative solutions. LR

finds applications in many physical problems, such as the inversion of light scattering data. In these problems, there are often additional information on the true solution that are usually ignored by many restoration methods because these quantities are likely to be affected by non negligible noise. In this chapter we propose a novel Weakly Constrained Lucy-Richardson (WCLR) method in which we add a weak constraint to the classical LR by introducing a penalization term, whose strength can be varied over a very large range. The WCLR method is simple and robust as the standard LR, but offers the great advantage of widely stretching the domain range over which the solution can be reliably recovered. Some selected numerical examples prove the performances of the proposed algorithm. The contents of this chapter are taken from [28].

Chapter 9. A semi-blind regularization algorithm In many inverse problems the operator to be inverted depends on a parameter which is not known precisely. In this chapter, following the idea from [17, 18], we propose a Tikhonov-type functional that involves as variables both the solution of the problem and the parameter on which the operator depends. We first prove that the non-convex functional admits a global minimum and that its minimization naturally leads to a regularization method. Then, using the popular Alternating Direction Multiplier Method (ADMM), we describe an algorithm to identify a stationary point of the functional. The introduction of the ADMM algorithm let us easily introduce some constraints on the reconstructions like nonnegativity and flux conservation. Since the functional is non-convex a proof of convergence of the method is given. Numerical examples prove the validity of the proposed approach. The contents of this chapter are taken from [29].

We conclude this introduction by remarking that the theoretical analysis in Chapters 3, 4, 7, and 8 is performed in the finite dimensional space $\mathbb{R}^n$ whereas in Chapters 5, 6, and 9 we study the problems at hand in the more general framework of infinite dimensional spaces.

In Figure 1.1 we propose a scheme of the structure of the thesis. We show the dependency between the chapters and the papers that correspond to each of it.

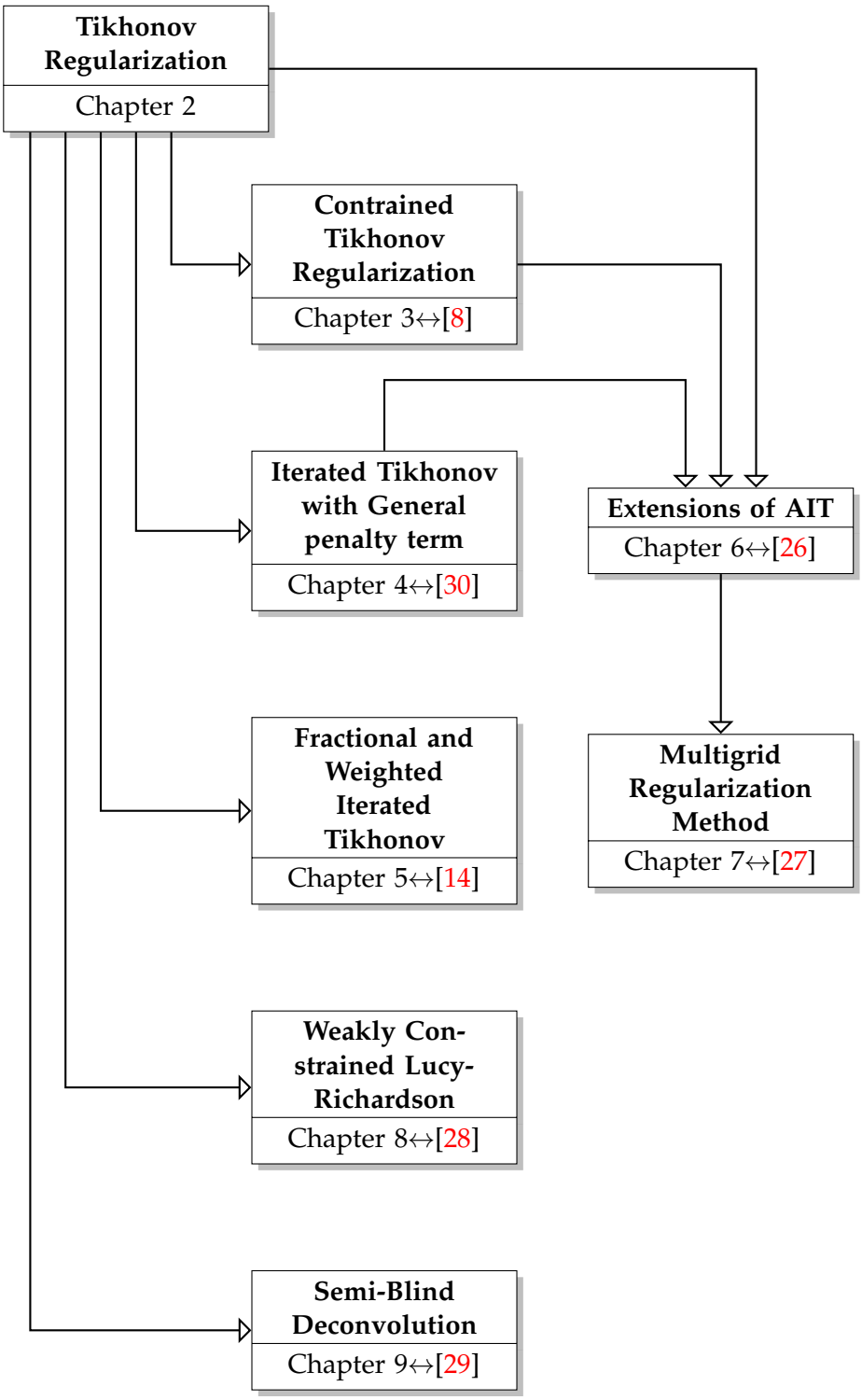


FIGURE 1.1: Schematic of the thesis

Chapter 2

Background

In this chapter we describe the type of problem we are interested in, moreover, we give some insight on Tikhonov regularization. This method is the keystone of all this thesis from which most of the work was derived.

2.1 Ill-posed problems

Definition 2.1. *We say that a mathematical problem is* well-posed *if*

(i) a solution exists;

(ii) the solution is unique;

(iii) the solution depends continuously on the data.

We say that a mathematical problem is ill-posed *if at least one of the conditions above does not hold.*

See [61] for a discussion on inverse problems.

An example of ill-posed problem are Fredholm integrals of the first kind

$$g(t) = \int_{\Omega} k(t,s) f(s) \mathrm{d}t, \tag{2.1}$$

here g denotes the available data, k is the integral kernel with compact support, and f is the signal we would like to recover.

Since k has compact support equation (2.1) is ill-posed. In fact, the solution does not depend continuously on the data.

When we discretize (2.1) we obtain a linear system

$$A\mathbf{x} = \mathbf{b},$$

where A is of ill-determined rank, i.e., its singular values decay gradually to zero without a significant gap. Least-squares problems with a matrix of this kind are commonly referred to as discrete ill-posed problems, see [61, 82] for discussions on ill-posed and discrete ill-posed problems.

The process of discretization, along with measurements errors and other factors, introduces noise inside the data $\mathbf{b}$, so that we have only access to $\mathbf{b}^{\delta}$ such that

$$\left\| \mathbf{b} - \mathbf{b}^{\delta} \right\| \leq \delta, \tag{2.2}$$

where $\|\cdot\|$ is the Euclidean norm and $\delta > 0$.

We can then formulate a linear least-squares problem of the form

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|A\mathbf{x} - \mathbf{b}^\delta\|, \quad A \in \mathbb{R}^{m\times n}, \quad \mathbf{b}^\delta \in \mathbb{R}^m. \tag{2.3}$$

Consider now the singular value decomposition (SVD) of A

$$A = U\Sigma V^t,$$

where $U \in \mathbb{R}^{m\times m}, V \in \mathbb{R}^{n\times n}$ are orthogonal matrices, $\Sigma \in \mathbb{R}^{m\times n}$ is a diagonal matrix whose diagonal entries σ_j are nonnegative and ordered in a decreasing way, and by A^t we denote the transpose of A. A is severely ill-conditioned and thus the singular values σ_j decreases to 0 very fast and with no gap.

The minimum norm solution of (2.3) can be obtained using the Moore-Penrose pseudo-inverse

$$A^\dagger = V\Sigma^\dagger U^t,$$

where by $\Sigma^\dagger$ we denote the $n \times m$ diagonal matrix whose diagonal elements are $\frac{1}{\sigma_j}$ for $\sigma_j \neq 0$ and zero otherwise.

Computing explicitly the solution of (2.3), assuming without loss of generality that $m > n$, we have

$$\mathbf{x}_{\text{naive}} = A^\dagger \mathbf{b}^\delta = \sum_{j=1}^{n} \frac{\mathbf{u}_j^t \mathbf{b}^\delta}{\sigma_j} \mathbf{v}_j.$$

Calling $\boldsymbol{\eta} = \mathbf{b}^\delta - \mathbf{b}$, we get

$$\mathbf{x}_{\text{naive}} = \sum_{j=1}^{n} \frac{\mathbf{u}_j^t \mathbf{b}}{\sigma_j} \mathbf{v}_j + \sum_{j=1}^{n} \frac{\mathbf{u}_j^t \boldsymbol{\eta}}{\sigma_j} \mathbf{v}_j.$$

Assuming that $\mathbf{b} \in \mathcal{R}(A)$, we have that

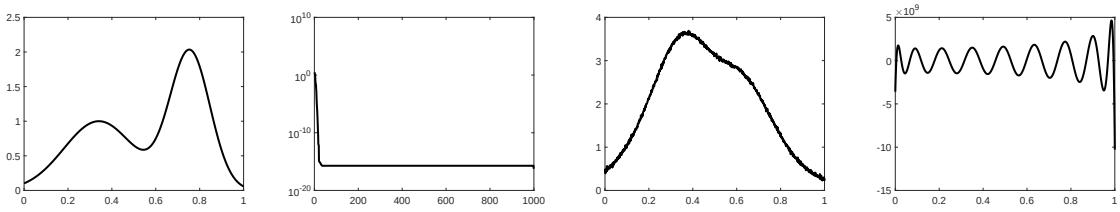
$$\frac{\mathbf{u}_j^t \mathbf{b}}{\sigma_j} \mathbf{v}_j = \frac{\mathbf{u}_j^t A\mathbf{x}}{\sigma_j} \mathbf{v}_j = \frac{\mathbf{u}_j U\Sigma V^t \mathbf{x}}{\sigma_j} \mathbf{v}_j = \frac{\sigma_j V^t \mathbf{x}}{\sigma_j} \mathbf{v}_j = V^t \mathbf{x} \mathbf{v}_j,$$

which does not depend on σ_j.

On the other hand, $\boldsymbol{\eta} \notin \mathcal{R}(A)$ and can be assumed as random. The singular vectors with big indexes are related to high frequencies and $\boldsymbol{\eta}$ will have non trivial components in this space. Since $\frac{1}{\sigma_j}$ becomes very large for j big enough the noise is amplified and completely corrupts the reconstruction.

This shows us that it is impossible to recover the original signal from $\mathbf{b}^\delta$, our task will be to formulate numerical methods that are able to provide good approximation of the true signal.

In Figure 2.1 we can see an example of ill-posed inverse problem. This is the `shaw` problem taken from the toolbox [83]. We have set $n = m = 1000$ and have added white Gaussian noise to the right-hand side such that $\delta = 0.01\,\|\mathbf{b}\|$. We can see that singular values of A decreases very fast with no gap and that the naive reconstruction is completely useless.



(a) (b) (c) (d)

FIGURE 2.1: `Shaw` test problem ($n = m = 1000$): (a) True signal, (b) Singular values, (c) Right-hand side $\mathbf{b}^\delta$ with $\delta = 0.01\,\|\mathbf{b}\|$, (d) Naive reconstruction $\mathbf{x}_{\text{naive}} = A^\dagger \mathbf{b}^\delta$.

2.1.1 Image Deblurring

We now move to describe one of the main ill-posed problem we are going to deal with: *image deblurring*. For a discussion on inverse problems in imaging refer to [13].

This inverse problem consists in recovering an image from a blurred and noisy version of itself. The blurring phenomenon can be modeled as a Fredholm integral of the first kind

$$g(s,t) = \int_\Omega k(s,u,t,v) f(u,v) \mathrm{d}u\mathrm{d}v, \tag{2.4}$$

where f is the true image, g is the measured data, and k is the blurring kernel. Usually we refer to k as to Point Spread Function (PSF) since it models how a single point is spread across its neighborhood.

We assume that the PSF has compact support, i.e., the blur in a point depends only on some pixels around it. The function k has nonnegative values and it holds

$$\int_\Omega k(s,u,t,v) \mathrm{d}s\mathrm{d}u\mathrm{d}t\mathrm{d}v = 1,$$

which means that it does neither create nor destroy information.

From this assumption on the PSF it is easy to see that

$$\int_\Omega f(s,t) \mathrm{d}s\mathrm{d}t = \int_\Omega g(s,t) \mathrm{d}s\mathrm{d}t,$$

i.e., the total mass (or, in this case, intensity of light), is preserved.

When the blur does not depend on the location, i.e., the PSF is the same in all the areas of the image, equation (2.4) reduces to

$$g(s,t) = \int_\Omega k(s-u, t-v) f(u,v) \mathrm{d}u\mathrm{d}v = k * f, \tag{2.5}$$

which is a convolution.

We have to discretize (2.5). When doing so we need to keep into account that we may not have access to all the domain Ω, but only to a small portion of it: the FOV. What is outside the FOV, however, has an impact on the blurred data we measure, but we do not have complete information on it. In other words, not knowing what is outside the FOV, we have to deal with an under-determined system, i.e., $n > m$.

There is, however, another possibility which then lead to a square $n \times n$ system, which are: *boundary conditions*, see [84].

Boundary Conditions

By boundary conditions we mean that we make assumptions on what is outside the FOV using the information that we already have inside. This assumptions have an effect on the structure of the matrix which can then be exploited to achieve fast computations.

We denote by $\mathbf{X}$ the true image inside the FOV, which is a matrix.

Zero The *zero* boundary condition is obtained by assuming that outside the FOV the image is 0 everywhere

$$\begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}.$$

This assumption is useful when dealing with astronomical images, since most of the time it is possible to assume that the outside the FOV the image is black, i.e., 0.

This choice leads to a blurring matrix A which is a block Toeplitz with Toeplitz block (BTTB). Unfortunately there are no fast transformation to diagonalize a general matrix of this form. We remind that a Toeplitz matrix is a matrix that is constant on the diagonals.

Periodic Another boundary condition is the *periodic*. We suppose that outside the FOV the image repeats itself in all directions

$$\begin{pmatrix} \mathbf{X} & \mathbf{X} & \mathbf{X} \\ \mathbf{X} & \mathbf{X} & \mathbf{X} \\ \mathbf{X} & \mathbf{X} & \mathbf{X} \end{pmatrix}.$$

This boundary condition does not ensure continuity on the boundaries and often leads to poor reconstructions.

In this case the blurring matrix A is block circulant with circulant blocks (BCCB) and it is diagonalized by the two dimensional Fourier matrix F that is constructed as follows. Consider the one dimensional Fourier matrix $F_1 \in \mathbb{C}^{n\times n}$ defined as

$$(F_1)_{j,k} = e^{-2(j-1)(k-1)\mathrm{i}\pi/n}, \quad \mathrm{i}^2 = -1. \tag{2.6}$$

The two dimensional Fourier matrix F is then defined as

$$F = F_1 \otimes F_1, \tag{2.7}$$

where $\otimes$ denotes the Kronecker product.

Reflective In the *reflective* case we assume that the image is reflected (like in a mirror) outside the FOV so that the image is continuous on the boundaries

$$\begin{pmatrix} \mathbf{X}_{\mathrm{x}} & \mathbf{X}_{\mathrm{ud}} & \mathbf{X}_{\mathrm{x}} \\ \mathbf{X}_{\mathrm{lr}} & \mathbf{X} & \mathbf{X}_{\mathrm{lr}} \\ \mathbf{X}_{\mathrm{x}} & \mathbf{X}_{\mathrm{ud}} & \mathbf{X}_{\mathrm{x}} \end{pmatrix},$$

where $\mathbf{X}_{\mathrm{ud}}$ is obtained by flipping the rows of $\mathbf{X}$, $\mathbf{X}_{\mathrm{lr}}$ is obtained by flipping the columns of $\mathbf{X}$ and $\mathbf{X}_{\mathrm{x}}$ is obtained by flipping both the columns and the rows of $\mathbf{X}$. Reflective boundary conditions where introduced in [105].

The resulting matrix A is a BTTB+BTHB+BHTB+BHHB matrix where by BTHB we denote a block Toeplitz with Hankel blocks, by BHTB we denote a block Hankel with Toeplitz blocks, and by BHHB we denote a block Hankel with Hankel blocks. We recall that a Hankel is a matrix which is constant on the anti-diagonals. This type of matrices can be diagonalized by the *discrete cosine transform* if the PSF is quadrantally symmetric, i.e., if the PSF is symmetric with regard to both the horizontal and vertical axes

Antireflective When we use the *antireflective* boundary conditions we are ensuring that on the boundary the image is not only continuous but is continuous also its normal derivative. In this case we antireflect the image outside the FOV. Indexing $\mathbf{X}$ inside the FOV as $(\mathbf{X})_{i,j} = X_{i,j}$ $i,j = 1,\ldots,n$ the extended image $\mathbf{X}_{\mathrm{e}}$ by $i,j = 1-p,\ldots,n+p$, where $p = n-m$. We obtain for the edges

$$\begin{aligned}
&\mathbf{X}_{\mathrm{e}}(1-i,j) = 2\mathbf{X}(1,j) - \mathbf{X}(i+1,j), \quad 1 \le p, 1 \le j \le n;\\
&\mathbf{X}_{\mathrm{e}}(i,1-j) = 2\mathbf{X}(i,1) - \mathbf{X}(i,j+1), \quad 1 \le n, 1 \le j \le p;\\
&\mathbf{X}_{\mathrm{e}}(n+1,j) = 2\mathbf{X}(n,j) - \mathbf{X}(n-1,j), \quad 1 \le p, 1 \le j \le n;\\
&\mathbf{X}_{\mathrm{e}}(i,n+j) = 2\mathbf{X}(i,n) - \mathbf{X}(i,n-j), \quad 1 \le n, 1 \le j \le p,
\end{aligned}$$

for the corners, i.e., when $1 \le i,j \le p$,

$$\begin{aligned}
&\mathbf{X}_{\mathrm{e}}(1-i,1-j) = 4\mathbf{X}(1,1) - 2\mathbf{X}(1,j+1) - 2\mathbf{X}(i+1,1) + \mathbf{X}(i+1,j+1);\\
&\mathbf{X}_{\mathrm{e}}(1-i,n+j) = 4\mathbf{X}(1,n) - 2\mathbf{X}(1,n-j) - 2\mathbf{X}(i+1,n) + \mathbf{X}(i+1,n-j);\\
&\mathbf{X}_{\mathrm{e}}(n+i,1-j) = 4\mathbf{X}(n,1) - 2\mathbf{X}(n,j+1) - 2\mathbf{X}(n-i,1) + \mathbf{X}(n-i,j+1);\\
&\mathbf{X}_{e}(n+i,n+j) = 4\mathbf{X}(n,n) - 2\mathbf{X}(n,n-j) - 2\mathbf{X}(n-i,n) + \mathbf{X}(n-i,n-j).
\end{aligned}$$

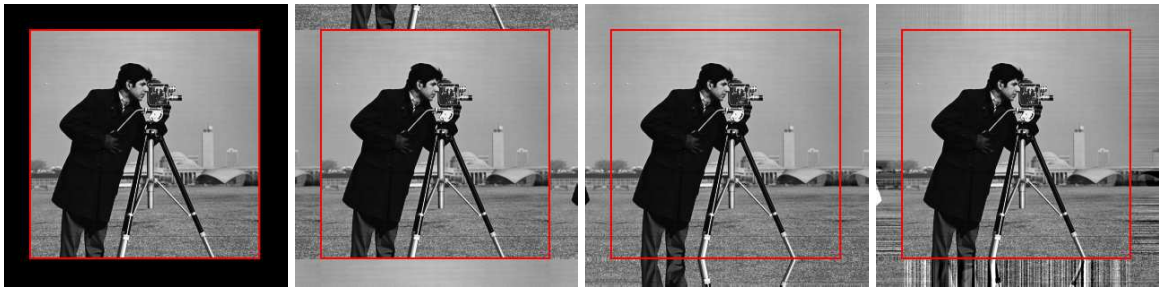
The structure of the resulting matrix A is quite complicated however, if the PSF is quadrantally symmetric, it can be diagonalized by a modification of the *discrete sine transform* [5, 47, 119].

In Figure 2.2 we show an example of the above described boundary conditions.

2.2 Tikhonov Regularization

Since directly solving an ill-posed problem is not possible, we have to resort to regularization methods. The regularized version of an ill-posed problem is a well-posed problem whose solution is an approximation of the desired solution

$$\mathbf{x}^{\dagger} = A^{\dagger}\mathbf{b}.$$



(a) (b) (c) (d)

FIGURE 2.2: Examples of boundary conditions, the red box delimits the FOV: (a) zero, (b) periodic, (c) reflective, (d) antireflective.

One of the most popular regularization method is Tikhonov regularization.

$$\mathbf{x}_\alpha = \arg\min_{\mathbf{x}\in\mathbb{R}^n} \left\| A\mathbf{x} - \mathbf{b}^\delta \right\|^2 + \alpha \left\| L\mathbf{x} \right\|^2, \tag{2.8}$$

where $\alpha > 0$ is the regularization parameter and $L \in \mathbb{R}^{q\times n}$ is the regularization operator, for a discussion on Tikhonov regularization, see [53, 61, 68, 75, 80, 82, 112]. Tikhonov regularization in (2.8) is called in *general form*. In order to have a unique solution we assume that

$$\mathcal{N}(A) \cap \mathcal{N}(L) = \{\mathbf{0}\}, \tag{2.9}$$

so that the solution of (2.8) can be computed by

$$\mathbf{x}_\alpha = \left(A^t A + \alpha L^t L\right)^{-1} A^t \mathbf{b}^\delta.$$

When we set $L = I$ we obtain Tikhonov regularization in *standard form*

$$\mathbf{x}_\alpha = \arg\min_{\mathbf{x}\in\mathbb{R}^n} \left\| A\mathbf{x} - \mathbf{b}^\delta \right\|^2 + \alpha \left\| \mathbf{x} \right\|^2, \tag{2.10}$$

note that in this case the condition (2.9) is trivially satisfied. The normal equations related to (2.10) are

$$\mathbf{x}_\alpha = (A^t A + \alpha I)^{-1} A^t \mathbf{b}^\delta. \tag{2.11}$$

The first term of Tikhonov regularization is a *data fitting* term and ensures that the reconstruction $\mathbf{x}_\alpha$ fits the measured data $\mathbf{b}^\delta$. The second term is called *penalty term* and requires that $\mathbf{x}_\alpha$ is smooth. The regularization operator L weights the norm in the penalty term so that some features of $\mathbf{x}$ are enhanced while other are penalized. Finally, the regularization parameter α balances the trade-off between the two terms. The determination of a good α is very important and can be tricky. If α is chosen too small, then the first term will prevail and the noise will corrupt the reconstruction. Instead, if α is too big, the second term will have more importance and the obtained approximation will be over-smoothed.

Many strategies have been proposed for the choice of both L and α [46, 52, 53, 64, 111]. One of the most popular rule for the choice of α, when δ is known, is the *discrepancy principle*. The parameter α is chosen so that

$$\left\| \mathbf{b}^\delta - A\mathbf{x}_\alpha \right\| = \tau\delta, \tag{2.12}$$

Method	Optimal α	RRE
Tikhonov in standard form	0.012742	0.10061
Tikhonov in general form	0.023357	**0.077164**

TABLE 2.1: Peppers test problem RRE comparison. In bold we highlight the best error.

where $\tau > 1$ is a constant. This criterion is based on the following observation: if $\mathbf{b} \in \mathcal{R}(A)$, then it holds

$$\left\|\mathbf{b}^\delta - A\mathbf{x}^\dagger\right\| = \left\|\mathbf{b}^\delta - \mathbf{b}\right\| \leq \delta.$$

2.2.1 Tikhonov in general form

As we stated above, usually, the quality of the reconstruction can be improved by switching from the standard form to the general form. Before analyzing the theoretical properties of Tikhonov regularization in general form we want to give an example. We consider an image deblurring test case. We start with the peppers image in Figure 2.3(a) and we blur it using the motion PSF in Figure 2.3(b). In this way we are simulating the effect obtained when a picture is taken with a camera that is moving. We then add white Gaussian noise such that $\delta = 0.02\,\|\mathbf{b}\|$ and obtain the resulting blurred and noisy image in Figure 2.3(c) We refer to the ratio

$$\xi = \frac{\|\boldsymbol{\eta}\|}{\|\mathbf{b}\|}, \tag{2.13}$$

as *noise level*. For the sake of simplicity in this example we are not considering the limited FOV and we are assuming that the true image is periodic.

We then reconstruct with Tikhonov in standard and general form. For the general form we use as regularization operator the discretization of the two dimensional divergence operator. Let L_1 be the finite difference discretization of the one dimensional derivative with periodic boundary conditions, i.e.,

$$L_1 = \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \\ 1 & & & & -1 \end{pmatrix},$$

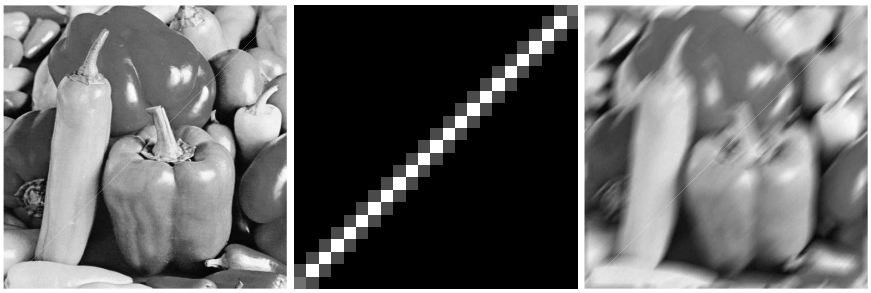
we define L as

$$L = L_1 \otimes I + I \otimes L_1, \tag{2.14}$$

By choosing the optimal α, i.e., the one that minimizes the error, we obtain the reconstruction in Figure 2.4. For the comparison of the two methods we consider the Relative Reconstruction Error (RRE) defined as

$$\text{RRE}(\mathbf{x}) = \frac{\left\|\mathbf{x} - \mathbf{x}^\dagger\right\|}{\|\mathbf{x}^\dagger\|} \tag{2.15}$$

In Table 2.1 we show the RRE obtained with the different methods. We can see that the introduction of the L is able to increase the accuracy of the method. Moreover, from the visual inspection of the reconstructions in Figure 2.4 we can see that Tikhonov in general form provides better restorations and in particular is able to lessen the so called ringing effect.



(a) (b) (c)

FIGURE 2.3: Peppers test problem: (a) True image (512×512 pixels), (b) motion PSF (23×23 pixels), (c) blurred and noisy image ($\xi = 0.02$).

(a) (b)

FIGURE 2.4: Peppers test problem reconstruction: (a) Tikhonov in standard form, (b) Tikhonov in general form. For both method the optimal α has been chosen.

We can now move to the theoretical analysis of (2.8). In particular, we want to see how it is possible to reduce the problem (2.8) in standard form (2.10).

If L is invertible, then the minimization problem (2.8) becomes

$$\min_{\overline{\mathbf{x}}=L\mathbf{x}} \left\| AL^{-1}\overline{\mathbf{x}} - \mathbf{b}^{\delta} \right\|^2 + \alpha \left\| \overline{\mathbf{x}} \right\|^2 . \tag{2.16}$$

Solving (2.16) leads to $\overline{\mathbf{x}}_\alpha$ from which we can retrieve the solution $\mathbf{x}_\alpha$ of (2.8) by multiplying times L^{-1}:

$$\mathbf{x}_\alpha = L^{-1}\overline{\mathbf{x}}_\alpha.$$

When L is not invertible, we follow [59]. Let $A : \mathcal{X} \to \mathcal{Y}$ and $L : \mathcal{X} \to \mathcal{X}$ be two linear operators between Hilbert spaces, the $A-$weighted pseudo-inverse of L is

$$L_A^{\dagger} = (I - (A(I - L^{\dagger}L))^{\dagger}A)L^{\dagger}. \tag{2.17}$$

We define the vectors

$$\begin{cases} \overline{\mathbf{x}} = L\mathbf{x} \\ \mathbf{x}^{(0)} = (A(I - L^{\dagger}L))^{\dagger}\mathbf{b}^{\delta} \\ \overline{\mathbf{b}}^{\delta} = \mathbf{b}^{\delta} - A\mathbf{x}^{(0)} \end{cases}$$

and consider the problem

$$\bar{\mathbf{x}}_\alpha = \arg\min_{\overline{\mathbf{x}}} \left\| AL_A^{\dagger}\overline{\mathbf{x}} - \mathbf{b}^{\delta} \right\|^2 + \alpha \left\| \overline{\mathbf{x}} \right\|^2 . \tag{2.18}$$

The solution $\mathbf{x}_\alpha$ of (2.10) is obtained from the solution $\overline{\mathbf{x}}_\alpha$ of (2.18) by

$$\mathbf{x}_\alpha = L_A^\dagger \overline{\mathbf{x}}_\alpha + \mathbf{x}^{(0)}.$$

2.2.2 Iterated Tikhonov

In order to furtherly improve the quality of the solution provided by Tikhonov regularization, it is possible to use a refinement technique and formulate an iterative algorithm.

The general idea behind refinement techniques is the following. Given the approximation $\mathbf{x}_k$ of $\mathbf{x}^\dagger$. Call

$$\mathbf{e}_k = \mathbf{x}^\dagger - \mathbf{x}_k,$$

the approximation error of $\mathbf{x}_k$, if we had access to $\mathbf{e}_k$ we could easily obtain $\mathbf{x}^\dagger$ form $\mathbf{x}_k$, since

$$\mathbf{x}^\dagger = \mathbf{x}_k + \mathbf{e}_k.$$

We can compute $\mathbf{e}_k$ from the so called *error equation*

$$A\mathbf{e}_k = A\left(\mathbf{x}^\dagger - \mathbf{x}_k\right) = A\mathbf{x}^\dagger - A\mathbf{x}_k = \mathbf{b} - A\mathbf{x}_k.$$

However, the computation of $\mathbf{e}_k$ is not trivial. We have to consider that A is severely ill-conditioned and that we do not know $\mathbf{b}$, but only $\mathbf{b}^\delta$. In other words we only have access to the system

$$A\mathbf{e}_k \approx \mathbf{r}_k = \mathbf{b}^\delta - A\mathbf{x}_k.$$

Therefore, letting $\mathbf{h}_k$ be an approximation of $\mathbf{e}_k$, we can refine $\mathbf{x}_k$ by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k.$$

We still have to resort to regularization methods to obtain a good approximation of $\mathbf{e}_k$, i.e., $\mathbf{h}_k$. We then use Tikhonov regularization in standard form to compute this approximation

$$\mathbf{h}_k = \arg\min_{\mathbf{h}} \|A\mathbf{h} - \mathbf{r}_k\|^2 + \alpha \|\mathbf{h}\|^2.$$

Summarizing we have

Algorithm 2.1 (Iterated Tikhonov (IT))**.** *Consider the linear system* (2.3)*. Let* $\mathbf{x}_0$ *be an initial guess for* $\mathbf{x}^\dagger$ *and let* $\alpha > 0$ *be a constant.*

```
for k = 1, 2, ...
        r_k = b^δ − A x_k
        x_{k+1} = x_k + (A^t A + αI)^{-1} A^t r_k
end
```

In Algorithm 2.1 the parameter α is chosen once and for all. The main issue with this approach is that the choice of α is, again, crucial and may be difficult.

A more stable algorithm is obtained when the parameter α is changed at each iteration. In order to have convergence, however, the sequence of the parameter should meet some

conditions. A sufficient condition is that

$$\sum_{k=0}^{\infty} \alpha_k^{-1} = \infty, \quad \forall k \ \alpha_k > 0.$$

We can formulate then the following

Algorithm 2.2 (Nonstationary Iterated Tikhonov (IT$_{NS}$))**.** *Consider the linear system* (2.3)*. Let* $\mathbf{x}_0$ *be an initial guess for* $\mathbf{x}^\dagger$ *and let* $\{\alpha_k\}_k$ *be a sequence such that* $\forall k \ \alpha_k > 0$ *and* $\sum_{k=0}^{\infty} \alpha_k^{-1} = \infty$*.*

for $k = 1, 2, \ldots$

$\mathbf{r}_k = \mathbf{b}^\delta - A\mathbf{x}_k$

$\mathbf{x}_{k+1} = \mathbf{x}_k + \left(A^t A + \alpha_k I\right)^{-1} A^t \mathbf{r}_k$

end

The convergence of both algorithm is assured by

Theorem 2.2 ([25])**.** *Consider the linear system* (2.3)*. Let* $\{\alpha_k\}_k$ *be a sequence such that* $\forall k \ \alpha_k > 0$ *and* $\sum_{k=0}^{\infty} \alpha_k^{-1} = \infty$*. Then the iterates generated by IT*$_{NS}$ *converges to the solution of* (2.3) *which is nearest to* $\mathbf{x}_0$*. In particular, if* $\mathbf{x}_0 = \mathbf{0}$*, then* $\mathbf{x}_k \to A^\dagger \mathbf{b}^\delta$ *as* $k \to \infty$*.*

The regularization inside both IT and IT$_{NS}$ is obtained by stopping the iteration before convergence. In particular we can still apply the discrepancy principle, but in this case we use it as a stopping criterion, i.e., we continue the iterations until we reach the first iteration k^* such that

$$\|\mathbf{r}_{k^*}\| \geq \tau\delta \ \text{ and } \ \|\mathbf{r}_k\| < \tau\delta \ \text{ for } \ k = 0, 1, \ldots, k^* - 1, \tag{2.19}$$

with $\tau > 1$.

The choice of the parameter α_k is very important. A possible and popular solution is the geometric sequence

$$\alpha_k = \alpha_0 q^k, \tag{2.20}$$

where $\alpha_0 > 0$ and $0 < q < 1$. Note that this sequence satisfies the hypothesis of Theorem 2.2. This choice is studied in [25, 79].

We would like to stress that for both algorithms we have used the standard form of Tikhonov regularization, i.e., we have set $L = I$. We will see in Chapter 4 how to analyze the case of stationary and nonstationary iterated Tikhonov when a general L is used.

Chapter 3

Constrained Tikhonov Minimization

As we said above, due to the fact that many singular values of the matrix A cluster at the origin, the least-squares problem (2.3) may be numerically rank-deficient. Therefore, it is generally beneficial to impose constraints on the computed solution that the desired solution $\mathbf{x}^\dagger$ is known to satisfy.

For instance, in image restoration problems the entries of the vector $\mathbf{x}^\dagger$ represent pixel values of the image. Pixel values are nonnegative and, therefore, it is generally meaningful to solve the constraint minimization problem

$$\mathbf{x}_\alpha^+ = \arg\min_{\mathbf{x}\geq 0} \left\| A\mathbf{x} - \mathbf{b}^\delta \right\|^2 + \alpha \left\| \mathbf{x} \right\|^2 \tag{3.1}$$

instead of (2.10). Here $\mathbf{x} \geq 0$ is intended component-wise. In this chapter we are going to consider only Tikhonov regularization in standard form. A closed form of the solution $\mathbf{x}_\alpha^+$ generally is not available.

Let Ω denote the nonnegative cone, i.e.,

$$\Omega = \{\mathbf{x} \in \mathbb{R}^n : (\mathbf{x})_i \geq 0 \; 1 \leq i \leq n\}, \tag{3.2}$$

and let P_Ω be the orthogonal projector from $\mathbb{R}^n$ to Ω. Thus, we determine $P_\Omega(\mathbf{z})$ for $\mathbf{z} \in \mathbb{R}^n$ by setting all negative entries of $\mathbf{z}$ to zero. An approximation of $\mathbf{x}_\alpha^+$ is furnished by

$$\mathbf{x}_\alpha^\Omega = P_\Omega(\mathbf{x}_\alpha) = P_\Omega\left(\left(A^tA + \alpha I\right)^{-1} A^t\mathbf{b}^\delta\right). \tag{3.3}$$

When $\mathbf{x}^\dagger \geq 0$, the vector $\mathbf{x}_\alpha^\Omega$ generally is a better approximation of $\mathbf{x}^\dagger$ than $\mathbf{x}_\alpha$. However, typically $\mathbf{x}_\alpha^+$ is a much more accurate approximation of $\mathbf{x}^\dagger$ than $\mathbf{x}_\alpha^\Omega$.

We want now to discuss the solution of the constrained Tikhonov regularization problem (3.1) by the modulus-based iterative method described in [122]. In [122] is discussed the application of this kind of method to the solution of nonnegative constrained least-squares (NNLS) problems,

$$\min_{\mathbf{x}\geq 0} \left\| A\mathbf{x} - \mathbf{b}^\delta \right\| \tag{3.4}$$

with a matrix $A \in \mathbb{R}^{m\times n}$ that is either well-conditioned or ill-conditioned. Since we are considering the case in which the singular values of A cluster at the origin we are able to determine accurate approximate solutions of (3.1) in a Krylov subspace of, generally, fairly small dimension. This observation reduces the computational effort considerably.

We also consider the situation when A is a BCCB matrix. Modulus-based iterative methods for the NNLS problem (3.4) with a matrix with this type of structure can be solved efficiently by application of FFT.

We first review results about modulus-based iterative methods discussed in [7, 57, 76, 122]. We then apply a modulus-based iterative method to the solution of large-scale constrained Tikhonov regularization problems (3.1). These problems are reduced to small size by a Krylov subspace method. This reduction lessens the computational effort required for the solution of the constrained Tikhonov regularization problem considerably.

We now give some comments on related work on the computation of nonnegative approximate solutions of problem (3.4) with a large matrix A whose singular values cluster at the origin. The importance of being able to solve this kind of problem has spurred the development of a variety of methods. In [102] is described a curtailed steepest descent method that determines nonnegative solutions. Active set methods based on Tikhonov regularization are developed in [98, 101] and barrier methods for Tikhonov regularization are discussed in [35, 99, 115]. A discussion of many optimization methods, including active set and barrier methods, is provided in [106]. In our experience, it is beneficial to use methods that exploit special properties or structure of the matrix A. The fact that the matrix A can be approximated well by a matrix of low rank makes our Krylov subspace modulus-based method competitive with available Krylov subspace methods, because it only requires the computation of one Krylov subspace of modest dimension. This subspace then is used repeatedly. When A is a BCCB matrix, fast solution methods are based on the fact that matrix-vector products with A and the (pseudo-)inverse of A can be computed in only $\mathcal{O}(n \log n)$ arithmetic floating point operations (flops) with the aid of the FFT.

This chapter is organized as follows: Section 3.1 reviews results about modulus-based iterative methods discussed in [7, 57, 76, 122]. We apply a modulus-based iterative method to the solution of large-scale constrained Tikhonov regularization problem (3.1). These problems are reduced to small size by a Krylov subspace method. This reduction lessens the computational effort required for the solution of the constrained Tikhonov regularization problem considerably. In Section 3.3 we briefly discuss the Golub-Kahan bidiagonalization technique. Section 3.4 describes our Krylov subspace-based method for the solution of (3.1) and Section 3.5 contains a few computed examples. The latter section also illustrates how the BCCB structure of the matrix A can be exploited.

3.1 Reformulation of the problem

This section summarizes results discussed in [57, 76, 122] of interest for the solution methods of the present chapter. Other recent discussions on modulus-based iterative methods can be found in [7, 10] and references therein.

We reduce the constrained least-squares problem (3.4) to a linear complementarity problem, which we will solve by a modulus-based iterative method. The following result can be found in [42, Page 5, Definition 3.3.1 and Theorem 3.3.7]. It is also shown in [122, Theorem 2.1].

Theorem 3.1. *Let M be a symmetric positive semidefinite matrix. Then the nonnegative constrained quadratic programming problem,*

$$\min_{\mathbf{x}\geq 0}\left(\frac{1}{2}\mathbf{x}^t M\mathbf{x}+\mathbf{c}^t\mathbf{x}\right),$$

denoted by NNQP(M, c), is equivalent to the linear complementarity problem,

$$\mathbf{x} \geq 0, \quad M\mathbf{x} + \mathbf{c} \geq 0, \quad \textit{and} \quad \mathbf{x}^t (M\mathbf{x} + \mathbf{c}) = 0,$$

denoted by LCP($M, \mathbf{c}$).

The results below, shown in [57, 76, 122], are consequences of the above theorem.
Corollary 3.2. *Let $M \in \mathbb{R}^{n\times n}$ be symmetric and positive definite and let $\mathbf{c} \in \mathbb{R}^n$. Then the problems NNQP($M, \mathbf{c}$) and LCP($M, \mathbf{c}$) have the same unique solution.*
Corollary 3.3. *The NNLS problem* (3.4) *is equivalent to LCP($A^tA, -A^t\mathbf{b}$),*

$$\mathbf{x} \geq 0, \quad \mathbf{r} = A^tA\mathbf{x} - A^t\mathbf{b} \geq 0, \quad \textit{and} \quad \mathbf{x}^t\mathbf{r} = 0.$$

It has a unique solution when A is of full column rank.
Theorem 3.4. *Let D be a positive definite diagonal matrix and define for $\mathbf{y} = [y_1, y_2, \ldots, y_n]^t \in \mathbb{R}^n$ the vector $|\mathbf{y}| = [|y_1|, |y_2|, \ldots, |y_n|]^t \in \mathbb{R}^n$.*

(i) If $(\mathbf{x}, \mathbf{r})$ is a solution of LCP($A^tA, -A^t\mathbf{b}$), then $\mathbf{y} = (\mathbf{x} - D^{-1}\mathbf{r})/2$ satisfies

$$\left(D + A^tA\right)\mathbf{y} = \left(D - A^tA\right)|\mathbf{y}| + A^t\mathbf{b}. \tag{3.5}$$

(ii) If $\mathbf{y}$ satisfies (3.5)*, then*

$$\mathbf{x} = |\mathbf{y}| + \mathbf{y} \quad \textit{and} \quad \mathbf{r} = D(|\mathbf{y}| - \mathbf{y})$$

is a solution of LCP($A^tA, -A^t\mathbf{b}$).

Proof. The results can be shown using [7, Theorem 2.1]. □

From here on we will assume that the matrix A has full column rank. This requirement is satisfied by the matrix $\tilde{A}$ used in the following; see(3.16).

3.2 Modulus Method

Theorem 3.4, and in particular equation (3.5), suggest the fixed-point iteration

$$(D + A^tA)\mathbf{y}_{k+1} = (D - A^tA)|\mathbf{y}_k| + A^t\mathbf{b}, \tag{3.6}$$

which is the basis for the following algorithm.
Algorithm 3.1 (Modulus Method (MM))**.** *Let $\mathbf{y}_0 \in \mathbb{R}^n$ be an initial approximate solution of* (3.5) *and let D be a positive definite diagonal matrix.*

$\mathbf{x}_0 = \mathbf{y}_0 + |\mathbf{y}_0|$
`for` $k = 0, 1, 2, \ldots$
  $\mathbf{y}_{k+1} = (D + A^tA)^{-1}\left((D - A^tA)|\mathbf{y}_k| + A^t\mathbf{b}\right)$
  $\mathbf{x}_{k+1} = \mathbf{y}_{k+1} + |\mathbf{y}_{k+1}|$
`end`

This algorithm is a special case of the modulus-based matrix splitting iterative methods proposed in [7]. Its convergence was investigated in [122] based on the analysis for HSS methods [9]. The case of interest to us is when $D = \mu I_n$ with $\mu > 0$. This iterative method is analyzed

in [57, 76]. We discuss the convergence of the iterates $\mathbf{y}_k$ for completeness. Let $\mathbf{y}^*$ denote the solution of (3.5) for $D = \mu I_n$, Then

$$\mathbf{y}_{k+1} - \mathbf{y}^* = (\mu I_n + A^t A)^{-1}(\mu I_n - A^t A)(|\mathbf{y}_k| - |\mathbf{y}^*|)$$

and we obtain

$$\begin{aligned} \|\mathbf{y}_{k+1} - \mathbf{y}^*\| &\leq \left\|(\mu I_n + A^t A)^{-1}(\mu I_n - A^t A)\right\| \||\mathbf{y}_k| - |\mathbf{y}^*|\| \\ &\leq \left\|(\mu I_n + A^t A)^{-1}(\mu I_n - A^t A)\right\| \|\mathbf{y}_k - \mathbf{y}^*\|. \end{aligned}$$

The matrix $(\mu I_n + A^t A)^{-1}(\mu I_n - A^t A)$ is symmetric. Therefore,

$$\left\|(\mu I_n + A^t A)^{-1}(\mu I_n - A^t A)\right\| = \max_{\lambda_j \in \lambda(A^t A)} \left|\frac{\mu - \lambda_j}{\mu + \lambda_j}\right|, \tag{3.7}$$

where $\lambda(A^t A)$ denotes the spectrum of $A^t A$. Since A is of full rank, $\lambda_j > 0$ for all j and, therefore,

$$\left|\frac{\mu - \lambda_j}{\mu + \lambda_j}\right| < 1 \qquad \forall j.$$

Hence,

$$\left\|(\mu I_n + A^t A)^{-1}(\mu I_n - A^t A)\right\| < 1,$$

which shows convergence of the iterations (3.6) for $D = \mu I_n$ with $\mu > 0$. The rate of convergence generally increases when (3.7) decreases. Replacing $\lambda(A^t A)$ in the right-hand side of (3.7) by its convex hull gives an optimization problem whose solution can be easily determined,

$$\mu^* = \arg\min_{\mu \in \mathbb{R}} \left\{ \max_{\mu_{\min} \leq \mu \leq \mu_{\max}} \left|\frac{\mu - \lambda}{\mu + \lambda}\right| \right\} = \sqrt{\lambda_{\min} \lambda_{\max}}. \tag{3.8}$$

Here $\lambda_{\min}$ and $\lambda_{\max}$ denote the smallest and largest eigenvalues of $A^t A$, respectively. Thus, the relaxation parameter μ^* gives a near-optimal rate of convergence.

3.3 Golub-Kahan bidiagonalization

Before formulating our algorithm, we are going to recall some basic facts on Golub-Kahan bidiagonalization.

Golub-Kahan bidiagonalization algorithm applied to the matrix $C \in \mathbb{R}^{m \times n}$ produces the following factorization

$$V^t C U = B, \tag{3.9}$$

where $V \in \mathbb{R}^{m \times m}$ and $U \in \mathbb{R}^{n \times n}$ are orthogonal matrices and $B \in \mathbb{R}^{m \times n}$ is a bidiagonal matrix. For the moment we follow [73, Section 10.4.1] and so B is going to be upper bidiagonal. Assume, without loss of generality, that $m > n$, we write B as

$$B = \begin{pmatrix} \alpha_1 & \beta_1 & \dots & \dots & 0 \\ 0 & \alpha_2 & \beta_2 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & 0 & \alpha_{n-1} & \beta_{n-1} \\ 0 & \dots & \dots & 0 & \alpha_n \\ \hline & & \mathbf{0} & & \end{pmatrix}$$

It is possible to show that a factorization of the form (3.9) exists if C is of full column rank (see [73, Section 5.4.8]). Moreover, since C and B are orthogonally related, they have the same singular values.

From (3.9) it follows that

$$CU = VB \quad \text{and} \quad C^tV = UB^t.$$

Writing V, U in terms of their columns

$$V = [\mathbf{v}_1|\dots|\mathbf{v}_m] \qquad U = [\mathbf{u}_1|\dots|\mathbf{u}_n]$$

yields to

$$C\mathbf{u}_j = \alpha_j\mathbf{v}_j + \beta_{j-1}\mathbf{v}_{j-1}, \tag{3.10}$$
$$C^t\mathbf{v}_j = \alpha_j\mathbf{u}_j + \beta_j\mathbf{u}_{j+1} \tag{3.11}$$

for $1 \le j \le n$, with the notation $\beta_0\mathbf{v}_0 \equiv \mathbf{0}$ and $\beta_n\mathbf{u}_{n+1} \equiv \mathbf{0}$. Define

$$\mathbf{r}_j = A\mathbf{u}_j - \beta_{j-1}\mathbf{v}_{j-1}, \tag{3.12}$$
$$\mathbf{p}_j = A^t\mathbf{v}_j - \alpha_j\mathbf{u}_j. \tag{3.13}$$

Combining (3.10), (3.12) and the orthonormality of the vectors $\mathbf{v}_j$ we get

$$\alpha_j = \pm\|\mathbf{r}_j\|,$$
$$\mathbf{v}_j = \mathbf{r}_j/\alpha_j, \quad (\alpha_j \neq 0).$$

Similarly from (3.11), (3.13) we have

$$\beta_j = \pm\|\mathbf{p}_j\|,$$
$$\mathbf{u}_{j+1} = \mathbf{p}_j/\beta_j, \quad (\beta_j \neq 0).$$

Using this relation we get the following

Algorithm 3.2 (Golub-Kahan upper bidiagonalization)**.** *Let $C \in \mathbb{R}^{m\times n}$ with full column rank. Let $\mathbf{u}_0 \in \mathbb{R}^n$ be a vector of unitary norm. The following procedure computes the factorization* (3.9)*.*

$k = 0,\ \mathbf{p}_0 = \mathbf{u}_0,\ \beta_0 = 1,\ \mathbf{v}_0 = \mathbf{0}$
While $\beta_j \neq 0$
  $\mathbf{u}_{j+1} = \mathbf{p}_j/\beta_j$
  $k = k + 1$
  $\mathbf{r}_j = C\mathbf{u}_j - \beta_{j-1}\mathbf{v}_{j-1}$
  $\alpha_j = \|\mathbf{r}_j\|$
  $\mathbf{v}_j = \mathbf{r}_j/\alpha_j$
  $\mathbf{p}_j = C^t\mathbf{v}_j - \alpha_j\mathbf{u}_j$
  $\beta_j = \|\mathbf{p}_j\|$
end

The above algorithm can be stopped after ℓ steps for C that are not of full column rank as long as ℓ is small enough.

It can be shown that

$$\operatorname{span}\{\mathbf{u}_1, \dots, \mathbf{u}_\ell\} = \mathcal{K}_\ell(C^tC, \mathbf{u}_0)$$
$$\operatorname{span}\{\mathbf{v}_1, \dots, \mathbf{v}_\ell\} = \mathcal{K}_\ell(CC^t, C\mathbf{u}_0)$$

where by $\mathcal{K}_\ell(C, \mathbf{y})$ we denote the Krylov subspace of dimension ℓ related to the pair $\{C, \mathbf{y}\}$, defined as

$$\mathcal{K}_\ell(C, \mathbf{y}) = \text{span}\{\mathbf{y}, C\mathbf{y}, \ldots, C^{\ell-1}\mathbf{y}\}.$$

By applying Algorithm 3.2 to C^t we obtain the factorization

$$V^t C^t U = B,$$

equivalently

$$U^t C V = B^t, \tag{3.14}$$

where U, V are orthonormal matrices and B^t is lower bidiagonal.

Algorithm 3.3 (Golub-Kahan lower bidiagonalization)**.** *Let $C \in \mathbb{R}^{m\times n}$ with full column rank. Let $\mathbf{u}_0 \in \mathbb{R}^n$ be a vector of unitary norm. The following procedure computes the factorization* (3.14)*.*

$k = 0,\ \mathbf{p}_0 = \mathbf{u}_0,\ \beta_0 = 1,\ \mathbf{v}_0 = \mathbf{0}$
`While` $\beta_j \neq 0$
  $\mathbf{u}_{j+1} = \mathbf{p}_j/\beta_j$
  $k = k + 1$
  $\mathbf{r}_j = C^t\mathbf{u}_j - \beta_{j-1}\mathbf{v}_{j-1}$
  $\alpha_j = \|\mathbf{r}_j\|$
  $\mathbf{v}_j = \mathbf{r}_j/\alpha_j$
  $\mathbf{p}_j = C\mathbf{v}_j - \alpha_j\mathbf{u}_j$
  $\beta_j = \|\mathbf{p}_j\|$
`end`

It then follows that

$$\text{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_\ell\} = \mathcal{K}_\ell(CC^t, \mathbf{u}_0)$$
$$\text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_\ell\} = \mathcal{K}_\ell(C^tC, C^t\mathbf{u}_0)$$

3.4 Krylov subspace methods for nonnegative Tikhonov regularization

We now combine the MM algorithm described in Section 3.2 with the Golub-Kahan lower bidiagonalization in Section 3.3. We describe the application of the modulus-based iterative method to Tikhonov regularization with nonnegativity constraint (3.1). We discuss how the computational effort for large-scale problems can be reduced by using a Krylov subspace method with a fixed Krylov subspace. Finally, we comment on how to exploit the BCCB structure of A in image deblurring applications.

Application of $\ell \ll \min\{m, n\}$ steps of Golub-Kahan lower bidiagonalization, i.e., of Algorithm 3.3, to A with initial vector $\mathbf{u}_0 = \mathbf{b}^\delta/\|\mathbf{b}^\delta\|$ gives the decompositions

$$AV_\ell = U_{\ell+1}B_{\ell+1,\ell}, \qquad A^tU_\ell = V_\ell B_{\ell,\ell}^t, \tag{3.15}$$

where $U_{\ell+1} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, u_{\ell+1}] \in \mathbb{R}^{m\times(\ell+1)}$ and $V_\ell = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_\ell] \in \mathbb{R}^{n\times\ell}$ have orthonormal columns, $U_\ell \in \mathbb{R}^{m\times\ell}$ is made up of the first ℓ columns of $U_{\ell+1}$, $B_{\ell+1,\ell} \in \mathbb{R}^{(\ell+1)\times\ell}$ is lower bidiagonal with positive diagonal and subdiagonal entries, and $B_{\ell,\ell}$ is the leading $\ell \times \ell$ submatrix of $B_{\ell+1,\ell}$.

We assume for now that ℓ is chosen small enough so that the decompositions (3.15) with the stated properties exist. As stated in the previous Section we note that the columns of V_ℓ span the Krylov subspace $\mathcal{K}_\ell(A^tA, A^t\mathbf{b}^\delta)$.

We first rewrite the minimization problem (3.1)

$$\begin{aligned} & \min_{\mathbf{x}\geq 0}\left\{\left\|A\mathbf{x}-\mathbf{b}^\delta\right\|^2+\alpha\|\mathbf{x}\|^2\right\} \\ = & \min_{\mathbf{x}\geq 0}\left\|\begin{pmatrix} A \\ \sqrt{\alpha}I_n \end{pmatrix}\mathbf{x}-\begin{pmatrix} \mathbf{b}^\delta \\ \mathbf{0} \end{pmatrix}\right\|^2 \\ = & \min_{\mathbf{x}\geq 0}\left\|\tilde{A}\mathbf{x}-\tilde{\mathbf{b}}^\delta\right\|^2, \end{aligned} \tag{3.16}$$

where we assume that $\alpha > 0$. Then the matrix $\tilde{A} \in \mathbb{R}^{(m+n)\times n}$ is of full column rank and the minimization problem (3.16) satisfies the conditions in Section 3.2. Therefore, the iterates determined by Algorithm 3.1 will converge.

When the matrix A is large and without exploitable structure, the computations with Algorithm 3.1 with $D = \mu I_n$ may be expensive. In particular, factoring the matrix $\mu I_n + \tilde{A}^t\tilde{A}$ in order to solve the linear systems of equations with this matrix required by Algorithm 3.1 may be unattractive or infeasible. We are interested in trying to reduce the computational effort required for solving these linear systems of equations. One way to achieve this is to solve them by the conjugate gradient method. It is convenient to use the CGLS implementation [16]. This solution approach is discussed in [122] and also illustrated in Section 3.5.

We now describe an alternative way to reduce the computational effort. We first determine an initial reduction of A to a small bidiagonal matrix with the aid of Golub-Kahan lower bidiagonalization. The Krylov solution subspace generated by this reduction method then is reused for all linear systems of equations with the matrix $\mu I_n + \tilde{A}^t\tilde{A}$ that have to be solved.

Substituting $\mathbf{x} = V_\ell\mathbf{y}$, $\mathbf{y} \in \mathbb{R}^\ell$, into (2.10) and determining an approximate solution by a Galerkin method gives the equation

$$V_\ell^t(A^tA + \alpha I_n)V_\ell\mathbf{y} = V_\ell^tA^t\mathbf{b}^\delta,$$

which, with the aid of the decompositions (3.15), can be expressed as

$$(B_{\ell+1,\ell}^tB_{\ell+1,\ell} + \alpha I_\ell)\mathbf{y} = \boldsymbol{e}_1\left\|A^t\mathbf{b}^\delta\right\|. \tag{3.17}$$

Here and below $\boldsymbol{e}_j$ denotes the jth column of an identity matrix of appropriate order. The reduced Tikhonov equations (3.17) are the normal equations associated with the least-squares problem

$$\min_{\mathbf{y}\in\mathbb{R}^\ell}\left\|\begin{pmatrix} B_{\ell+1,\ell} \\ \sqrt{\alpha}I_\ell \end{pmatrix}\mathbf{y}-\sqrt{\alpha}\boldsymbol{e}_{\ell+2}\left\|A^t\mathbf{b}^\delta\right\|\right\|. \tag{3.18}$$

We solve the latter instead of (3.17) for $\mathbf{y} = \mathbf{y}_\alpha$ for reasons of numerical stability. For each fixed $\alpha > 0$ the least-squares problem (3.18) can be solved in only $\mathcal{O}(\ell)$ arithmetic floating-point operations [60] for details on the solution of least-squares problems of the form (3.18).

We turn to the determination of $\alpha > 0$ by the discrepancy principle. Substituting $\mathbf{x}_\alpha = V_\ell\mathbf{y}$ into (2.12) and using (3.15) gives the reduced problem

$$\|B_{\ell+1,\ell}\mathbf{y} - \boldsymbol{e}_1\|\mathbf{b}^\delta\|\,\| = \tau\delta, \tag{3.19}$$

where $\mathbf{y}$ solves (3.18).

Proposition 3.5. *Introduce the function*

$$\phi_\ell(\alpha) = \|\mathbf{b}^\delta\|^2 \boldsymbol{e}_1^t (\alpha^{-1} B_{\ell+1,\ell} B_{\ell+1,\ell}^t + I_{\ell+1})^{-2} \boldsymbol{e}_1. \tag{3.20}$$

Then the solution $\alpha > 0$ of

$$\phi_\ell(\alpha) = \tau^2 \delta^2 \tag{3.21}$$

determines a solution $\mathbf{y} = \mathbf{y}_\alpha$ of (3.18) *that solves* (3.19). *The vector $\mathbf{x}_{\alpha,\ell} = V_\ell \mathbf{y}_\alpha$ satisfies* (2.12).

Proof. It follows from (3.15) that

$$A^t \mathbf{b}^\delta = A^t U_\ell \boldsymbol{e}_1 1 \|\mathbf{b}^\delta\| = \mathbf{v}_1 \, \boldsymbol{e}_1^t B_{\ell+1,\ell}^t \boldsymbol{e}_1 \|\mathbf{b}^\delta\|.$$

Substituting this expression and the solution of (3.17) into the left-hand side of (3.19) gives

$$\begin{aligned} & \|B_{\ell+1,\ell}(B_{\ell+1,\ell}^t B_{\ell+1,\ell} + \alpha I_\ell)^{-1} \boldsymbol{e}_1 \|A^t \mathbf{b}^\delta\| - \boldsymbol{e}_1 \|\mathbf{b}^\delta\| \, \|^2 \\ = \; & \|B_{\ell+1,\ell}(B_{\ell+1,\ell}^t B_{\ell+1,\ell} + \alpha I_\ell)^{-1} B_{\ell+1,\ell}^t \boldsymbol{e}_1 - \boldsymbol{e}_1 \|^2 \, \|\mathbf{b}^\delta\|^2. \end{aligned} \tag{3.22}$$

The identity

$$B_{\ell+1,\ell}(B_{\ell+1,\ell}^t B_{\ell+1,\ell} + \alpha I_\ell)^{-1} B_{\ell+1,\ell}^t - I_{\ell+1} = -(\alpha^{-1} B_{\ell+1,\ell} B_{\ell+1,\ell}^t + I_{\ell+1})^{-1}$$

can be shown, e.g., by multiplication by $B_{\ell+1,\ell} B_{\ell+1,\ell}^t + \alpha I_{\ell+1}$ from the right-hand side. Substitution into (3.22) gives

$$\|B_{\ell+1,\ell} \mathbf{y}_\alpha - \boldsymbol{e}_1 \|\mathbf{b}^\delta\| \, \|^2 = \|\mathbf{b}^\delta\|^2 \boldsymbol{e}_1^t (\alpha^{-1} B_{\ell+1,\ell} B_{\ell+1,\ell}^t + I_{\ell+1})^{-2} \boldsymbol{e}_1,$$

This shows (3.20). The fact that the vector $\mathbf{x}_{\mu,\ell}$ satisfies (2.12) follows from (3.15) and (3.19). □

Proposition 3.6. *Let $\phi_\ell(\alpha)$ be defined by* (3.20). *Then the function $\nu \to \phi_\ell(1/\nu)$ is strictly decreasing and convex for $\nu > 0$. Moreover,*

$$\lim_{\alpha \to \infty} \phi_\ell(\alpha) = \|\mathbf{b}^\delta\|^2.$$

In particular, Newton's method applied to the solution of the equation $\phi_\ell(1/\nu) = \tau^2 \delta^2$ with initial approximate solution ν_0 to the left of the solution converges monotonically and quadratically.

Proof. The decrease, convexity, and limit follows from the representation

$$\phi_\ell(1/\nu) = \|\mathbf{b}^\delta\|^2 \boldsymbol{e}_1^t (\nu B_{\ell+1,\ell} B_{\ell+1,\ell}^t + I_{\ell+1})^{-2} \boldsymbol{e}_1.$$

Newton's method converges monotonically and quadratically for decreasing convex functions when the initial iterate is smaller than the solution. The initial iterate ν_0 can be chosen to be zero with

$$\lim_{\nu \searrow 0} \phi_\ell(1/\nu) = \|\mathbf{b}^\delta\|^2, \qquad \lim_{\nu \searrow 0} \frac{d}{d\nu} \phi_\ell(1/\nu) = -2\|\mathbf{b}^\delta\|^2 \|B_{\ell+1,\ell}^t \boldsymbol{e}_1\|^2.$$

□

In actual computations it typically suffices to choose $\ell \ll \min\{m, n\}$. We apply MM Algorithm 3.1 to the reduced Tikhonov minimization problem (3.17). Thus, we replace $A^t A$ in the

algorithm by

$$T_{\ell,\alpha} = B_{\ell+1,\ell}^t B_{\ell+1,\ell} + \alpha I_\ell. \tag{3.23}$$

Since the matrix $B_{\ell+1,\ell}$ is small, we can easily determine its largest singular value $\sigma_{\max}$. Typically, zero is a quite sharp lower bound for the smallest singular value. The largest eigenvalue of $T_{\ell,\alpha}$ is $\sigma_{\max}^2 + \mu$ and the smallest eigenvalue is bounded below by, and is generally close to, α. Hence, we will use the relaxation parameter

$$\mu = \sqrt{(\sigma_{\max}^2 + \alpha)\alpha} \tag{3.24}$$

for the algorithm; cf. (3.8). This yields the following scheme.

Algorithm 3.4 (Krylov subspace Modulus Method)**.** *Choose the number of Golub-Kahan lower bidiagonal steps, ℓ, and compute the decompositions* (3.15)*. Determine a regularization parameter α that satisfies* (3.21) *as described in Proposition 3.6. Compute the solution $y = y_\alpha$ of* (3.18) *and define the initial approximate solution $\mathbf{x}_0 = P_\Omega(V_\ell y_\alpha)$ of* (3.1)*. Determine the largest singular value of the matrix $B_{\ell+1,\ell}$ and define the relaxation parameter* (3.24)*. Let $T_{\ell,\alpha}$ be given by* (3.23)*.*

$$\begin{aligned}
&\hat{\mathbf{b}}^\delta = \boldsymbol{e}_1 \left\| A^t \mathbf{b}^\delta \right\| \\
&\mathbf{y}_0 = V_\ell^t \mathbf{x}_0 \\
&\tilde{\mathbf{y}}_0 = V_\ell^t |V_\ell \mathbf{y}_0| \\
&\texttt{for } k = 0, 1, 2, \ldots \texttt{ until convergence} \\
&\quad \mathbf{y}_{k+1} = (\mu I_\ell + T_{\ell,\alpha})^{-1} \left((\mu I_\ell - T_{\ell,\alpha}) \tilde{\mathbf{y}}_k + \hat{\mathbf{b}}^\delta \right) \\
&\quad \tilde{\mathbf{y}}_{k+1} = V_\ell^t |V_\ell \mathbf{y}_{k+1}| \\
&\texttt{end} \\
&\mathbf{x} = V_\ell \tilde{\mathbf{y}}_{k+1} + |V_\ell \tilde{\mathbf{y}}_{k+1}|
\end{aligned}$$

The above algorithm computes the magnitude of every entry of an n-vector at each step. Therefore, a transformation from the ℓ-dimensional subspace, where the vectors $\mathbf{y}_k$ live, to $\mathbb{R}^n$ is required. Every step demands the solution of a linear system of equations of the form

$$(\mu I_\ell + T_{\ell,\alpha})\, \mathbf{z} = \mathbf{d}$$

for some vector $\mathbf{d}$. The solution $\mathbf{z}$ can be computed by solving a least-squares problem analogous to (3.18).

We remark that the Krylov subspace $\mathcal{K}(A^tA, A^t\mathbf{b}^\delta)$ is invariant under shifts of A^tA by a multiple of the identity, i.e.,

$$\mathcal{K}_\ell(A^tA, A^t\mathbf{b}^\delta) = \mathcal{K}_\ell(A^tA + \mu I_n, A^t\mathbf{b}^\delta).$$

It follows that the shifted matrix $\mu I_\ell + T_{\ell,\alpha}$ in Algorithm 3.4 corresponds to the shifted matrix $\mu I_n + \left(A^tA + \alpha I_n\right)$.

We described above how to determine the regularization parameter α by first reducing equation (2.12) to an equation with a small matrix (3.19). When restoring images, we sometimes may impose periodic boundary conditions without affecting the quality of the computed restoration significantly. This yields a BCCB blurring matrix $A \in \mathbb{R}^{n\times n}$, which can be diagonalized by the unitary Fourier matrix F defined in (2.7),

$$A = F^*\Sigma F. \tag{3.25}$$

Here the matrix Σ is diagonal, possibly with complex entries, and the superscript * denotes transposition and complex conjugation; see, e.g., [84] for details. We can transform the Tikhonov minimization problem (2.10) to a minimization problem with a diagonal matrix, and we also can transform (2.12) to an equation with a diagonal matrix. These transformations allow easy computation of the regularization parameter $\alpha > 0$ such that the solution (2.10) satisfies (2.12) by Newton's method analogously as described above. Moreover, Algorithm 3.1 with $D = \mu I_n$ can be executed efficiently when A has the factorization (3.25). This is illustrated in the following section.

3.5 Numerical examples

This section presents a few numerical examples that illustrate the performance of the algorithms described. An example in one-dimensional space from the REGULARIZATION TOOLS MATLAB package [83] and examples in two-dimensional space obtained with the RESTORE TOOLS MATLAB package [12] will be discussed. We compare four methods: classical unconstrained Tikhonov regularization (2.10), projected Tikhonov regularization (3.3), and Algorithms 3.1 and 3.4. In Algorithm 3.1, we solve linear systems of equations with the matrix $D + A^tA$ by the CGLS method; see [16]. In all examples, $D = \mu I_n$ with $\mu > 0$. Algorithm 3.5 below illustrates how the availability of a factorization of the form (3.25) can be exploited.

At step k of Algorithm 3.1, we have to solve the linear system of equations

$$(A^tA + \alpha I_n + \mu I_n)\mathbf{y}_{k+1} = (\mu I_n - A^tA - \alpha I_n)|\mathbf{y}_k| + A^t\mathbf{b}^\delta, \tag{3.26}$$

which is equivalent to the least-squares problem

$$\mathbf{y}_{k+1} = \arg\min_{\mathbf{y}} \left\| \begin{pmatrix} A \\ \sqrt{\alpha} I_n \\ \sqrt{\mu} I_n \end{pmatrix} \mathbf{y} - \begin{pmatrix} \mathbf{b}^\delta - A|\mathbf{y}_k| \\ -\sqrt{\alpha}|\mathbf{y}_k| \\ \sqrt{\mu}|\mathbf{y}_k| \end{pmatrix} \right\|^2 = \arg\min_{\mathbf{y}} \left\| \bar{A}\mathbf{y} - \bar{\mathbf{y}}_k \right\|$$

for a suitably defined matrix $\bar{A} \in \mathbb{R}^{(m+2n)\times n}$ and vector $\bar{\mathbf{y}}_k \in \mathbb{R}^{(m+2n)}$. We terminate the iterations with the CGLS method at iteration k of Algorithm 3.1 as soon as

$$\left\| \bar{A}^t(\bar{A}\mathbf{y} - \bar{\mathbf{y}}_k) \right\| < \frac{10^{-2}}{k} \left\| \bar{A}^t\bar{\mathbf{y}}_k \right\|, \text{ for } k = 0, 1, \ldots.$$

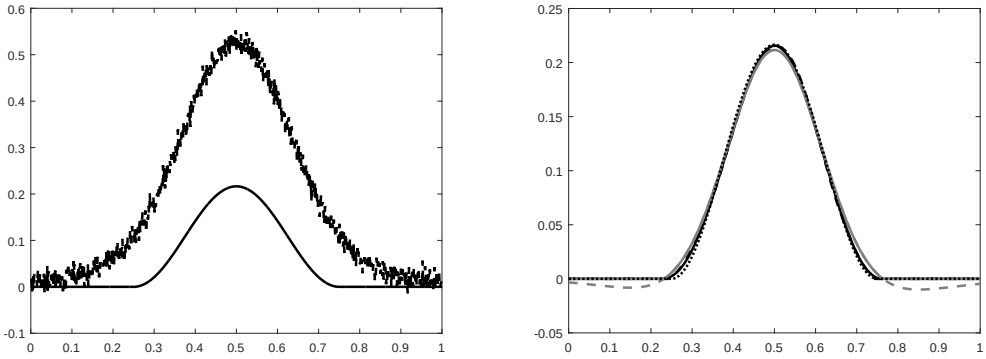
This stopping criterion takes the scalings of A and $\mathbf{b}^\delta$ into account. Both execution times and accuracy of the methods in our comparison are tabulated. The accuracy of a computed approximation $\mathbf{x}$ of $\mathbf{x}^\dagger$ is measured by the RRE defined in (2.15).

The iterations with Algorithms 3.1 and 3.4 are terminated when two consecutive iterates $\mathbf{y}_k$ and $\mathbf{y}_{k+1}$ are close enough, i.e., as soon as

$$\frac{\|\mathbf{y}_{k+1} - \mathbf{y}_k\|}{\|\mathbf{y}_k\|} < s,$$

where s is a user-supplied constant. We set $s = 10^{-4}$ in all examples.

The regularization parameter α is determined by the discrepancy principle, i.e., α is chosen such that (2.12) holds, with $\tau = 1.01$.



(a) (b)

FIGURE 3.1: `Shaw` test problem: (a) desired solution $\mathbf{x}^\dagger$ (solid curve) and error-contaminated data vector $\mathbf{b}^\delta$ (dashed curve), (b) computed solutions obtained with classical Tikhonov x_α (dashed gray curve), projected Tikhonov x_α^Ω (solid gray curve), Algorithm 3.1 (dashed black curve), and Algorithm 3.4 (solid black curve).

Determination of a near-optimal relaxation parameter μ for Algorithms 3.4 is straightforward, see the discussion preceding (3.24).

We turn to Algorithm 3.1. This algorithm requires the estimation of the largest and smallest eigenvalues of the matrix $A^tA + \alpha I_n$; see (3.26). We briefly comment on how these eigenvalues can be computed when the matrix A is large. Let A be scaled to have norm about unity. Since A stems from the discretization of an ill-posed problem, it has many singular values close to the origin. It follows that an accurate estimate of the smallest eigenvalue of the matrix $A^tA + \alpha I_n$ is given by α. An estimate of the largest eigenvalue of this matrix can be determined by computing an estimate of the largest singular value of A. This can be done quite inexpensively with the implicitly restarted Golub-Kahan bidiagonalization algorithm `irbla` described in [6]. The dominant computational work with this algorithm consists of the evaluation of a few matrix-vector products with the matrices A and A^t. The discussion and computed examples presented in [107] indicate that the `irbla` algorithm typically only requires the evaluation of a few of these matrix-vector products to determine the largest singular value of a matrix of a linear discrete ill-posed problem (2.3). The computation of an estimate of the largest singular value of A, and hence of the largest eigenvalue of $A^tA + \alpha I_n$, therefore is quite inexpensive.

In image restoration problems A is a blurring matrix. For a typical row j, blurring matrices satisfy $e_j^tA\mathbf{1} = 1$, where $\mathbf{1} = [1, 1, \ldots, 1]^t$. However, both $e_j^tA\mathbf{1}$ and $e_j^tA^tA\mathbf{1}$ may differ from one for certain j. In particular, $e_j^tA^tA\mathbf{1}$ may be significantly larger than one for some j values. The size of $\max_{1\leq j\leq n} |e_j^tA^tA\mathbf{1}|$ depends on the boundary conditions used; see, e.g., [50] for a discussion. We conclude that for some image restoration problems, the largest singular value of A is close to unity and does not have to be computed. However, certain image restoration problems, in particular problems with antireflective boundary conditions, may require that the largest singular value of the blurring matrix be computed as described above.

All the computations for this section were carried out in MATLAB version 9.0.0.341360 (R2016a) on a laptop computer with an Intel i7-6700HQ @ 2.60 Ghz CPU and 8 GB of RAM. The computations were done with about 15 significant decimal digits.

Method	RRE	CPU time	Iterations
Tikhonov	0.073600	0.39441	–
Projected Tikhonov	0.052816	0.40583	–
Algorithm 3.1	0.029923	0.61289	36
Algorithm 3.4	**0.024316**	0.062824	37

TABLE 3.1: `Shaw` test problem: relative errors (RRE) and CPU times in seconds for standard Tikhonov (2.10), projected Tikhonov (3.3), Algorithm 3.1, and Algorithm 3.4. For the last two methods also the number of iterations is displayed. The smallest error is shown in boldface.

Shaw We consider a modified version of the `shaw` example in [83]. To show the effectiveness of our method, we use the discretized integral operator from `shaw` and the exact solution $\mathbf{x}^\dagger$ from the `phillips` example, also from [83]. We choose this solution vector because it is nonnegative with many vanishing components. The discretized operator is represented by a matrix $A \in \mathbb{R}^{1024\times 1024}$. Thus, $\mathbf{x}^\dagger \in \mathbb{R}^{1024}$ represents the desired solution and the noise-free data vector is given by $\mathbf{b} = A\mathbf{x}^\dagger$. We add 5% white Gaussian noise to $\mathbf{b}$, i.e., we set $\xi = 0.05$ in (2.13), to obtain the noise-contaminated vector $\mathbf{b}^\delta$ in (2.3).

Figure 3.1(a) shows the vectors $\mathbf{x}^\dagger$ and $\mathbf{b}^\delta$. For Algorithm 3.4, we use a Krylov subspace of dimension $\ell = 30$. Table 3.1 displays CPU times as well as the relative errors in the computed approximations of $\mathbf{x}^\dagger$ determined by the different methods. Algorithm 3.4 is seen to require fewer iterations and less CPU time than the other methods. The standard Tikhonov method is implemented by first computing the singular value decomposition of A. We remark that Algorithm 3.4 performs particularly well for discrete ill-posed problems (2.3) with a matrix A whose singular values decay to zero fairly quickly, because in this situation the dimension ℓ of the Krylov subspace can be chosen fairly small. When the singular values decay slowly and, therefore, ℓ has to be chosen rather large, Algorithm 3.1 may be competitive with Algorithm 3.4.

We now compare Algorithm 3.4 with an active set method designed for the solution of nonnegatively constrained linear discrete ill-posed problems (3.4). Our comparison is with the method described in [101]. The performances of this active set method and the one discussed in [98] are quite similar. We therefore only compare with the former. It is based on repeatedly reducing the large problem (2.3) to a problem of small size with the aid of a few steps of Golub-Kahan bidiagonalization of the matrix A or of a matrix AD. Here D is a diagonal matrix with diagonal entries one or zero. The diagonal entries are zero if the corresponding variable is in the active set. The reduction of A or AD by Golub-Kahan bidiagonalization proceeds until an approximate solution that satisfies the discrepancy principle has been found. If the computed approximate solution satisfies the constraints, then we are done; otherwise those variables that violate the constraint are projected into the feasible set and the active set is updated. This means that the matrix D is updated. If the projected solution satisfies the discrepancy principle then we also are done; otherwise a partial Golub-Kahan bidiagonalization of the new matrix AD is computed. The computations proceed in this manner until a feasible approximate solution of (3.4) that satisfies the discrepancy principle has been found. Updating the active set only when the discrepancy principle holds gives a much faster method than if the active set were updated as soon as a constraint is violated. However, this updating strategy may allow "cycling". It is discussed in [98] how cycling can be detected and avoided. Computed examples in [98, 101] show that the active set methods to perform well when the noise level is not small. However, for small noise levels many partial Golub-Kahan bidiagonalizations may have to be computed. This requires the evaluation of many matrix-vector products (MvPs) and can make the methods slow. We remark that the

Noise Level	Method	RRE	MvPs
0.1%	Active set method	0.018188	36
	Algorithm 3.4	0.013495	30 ($\ell = 15$)
0.05%	Active set method	0.0093832	47
	Algorithm 3.4	0.013320	30 ($\ell = 15$)

TABLE 3.2: `Shaw` test problem: relative errors (RRE) and number of MvPs for the active set method [98] and for Algorithm 3.4. Results are shown for two noise levels. The smallest error is shown in boldface.

evaluation of these MvPs is the dominating work for large-scale problems.

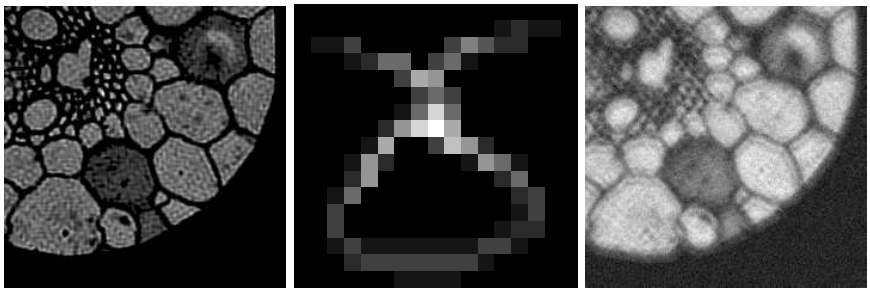
Table 3.2 illustrates that, differently from the active set method [101], Algorithm 3.4 does not require more computational effort when the noise level is reduced. We compare Algorithm 3.4 in terms of accuracy of the computed restoration and in terms of the number of MvPs evaluations required. For Algorithm 3.4 the number of MvPs needed depends only on the dimension of the Krylov solution subspace used. Since we use Golub-Kahan bidiagonalization, the computation of a solution subspace of dimension ℓ requires the evaluation of 2ℓ MvPs. Table 3.2 displays results for two noise levels. The table shows that when the noise level decreases the number of MvPs evaluations required by the active set method [101] increases, while it does not for Algorithm 3.4. In fact, it may be possible to choose a Krylov subspace of smaller dimension for Algorithm 3.4 for small noise levels and this would reduce the number of MvPs evaluations required.

Grain We turn to image deblurring in two-dimensional space. We blur the true image Grain from [12] using a non-symmetric PSF and add 10% white Gaussian noise; see Figure 3.2. Antireflective boundary conditions are imposed; see [48, 54, 119] for details. There is no fast transformation that can be applied to diagonalize the blurring matrix A. Therefore, we use Algorithm 3.4 to compute a restoration. We compare this algorithm to standard and projected Tikhonov regularization (2.10) and (3.3), respectively, and to Algorithm 3.1, in which the inner linear systems of equations are solved by the CGLS method; cf. the discussion at the beginning of Section 3.5. In Algorithm 3.4, we apply a Krylov subspace of dimension $\ell = 100$. Table 3.3 displays CPU times and the errors in the computed restorations determined by these methods. We see that Algorithm 3.1 requires about the same computing time as standard and projected Tikhonov regularization, and that Algorithm 3.4 is much faster than Algorithm 3.1. Moreover, Algorithm 3.4 gives the most accurate restoration. This is confirmed by visual inspection of Figure 3.3.

In this and the following examples, Tikhonov regularization (2.11) is implemented by solving the least-squares problem

$$\min_{\mathbf{y}\in\mathbb{R}^n} \left\| \begin{pmatrix} A \\ \sqrt{\alpha} I_n \end{pmatrix} \mathbf{y} - \begin{pmatrix} \mathbf{b}^\delta \\ \mathbf{0} \end{pmatrix} \right\|,$$

by the CGLS method. Here $\mathbf{0} \in \mathbb{R}^n$ denotes the zero vector. The α-value is determined as follows. Let C denote the blurring matrix obtained by using periodic boundary conditions. We compute α that satisfies (3.21), where we substitute the matrix $B_{\ell+1,\ell}$ in (3.20) by C and exploit the factorization (3.25). Proceeding in this way yields a suitable value of the regularization parameter α in a computationally efficient manner. We remark that we are primarily interested in the errors in the solutions determined by the different methods. Therefore, it is not necessary to implement the standard Tikhonov method as a black box method.

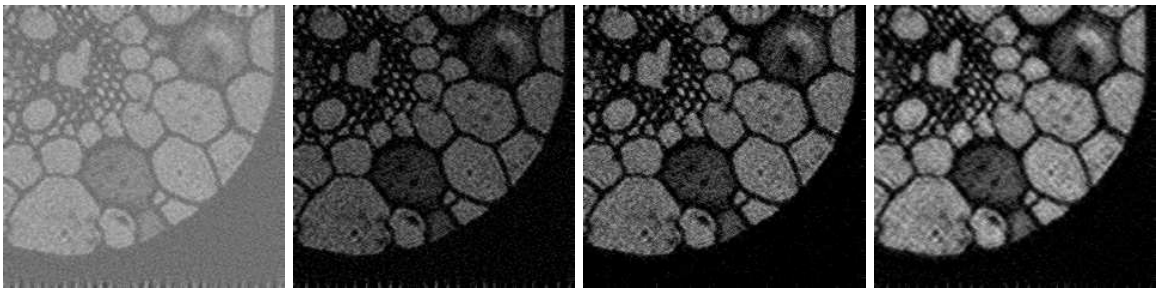


(a) (b) (c)

FIGURE 3.2: Grain test problem: (a) true image (238×238 pixels), (b) non-symmetric PSF (17×17 pixels), (c) blurred and noisy image.

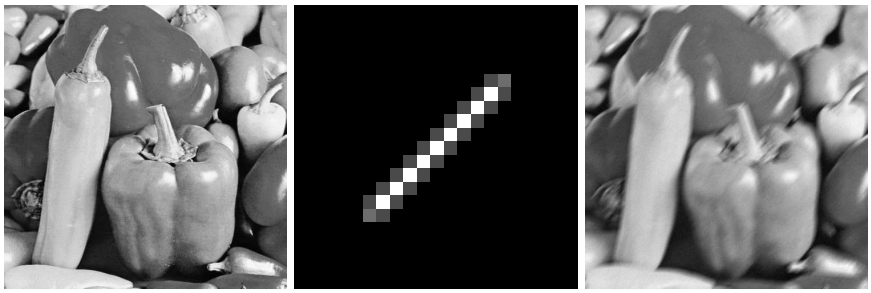
Method	RRE	CPU time	Iterations
Tikhonov	0.33043	9.2475	–
Projected Tikhonov	0.30239	9.2726	–
Algorithm 3.1	0.27633	8.1002	24
Algorithm 3.4	**0.27491**	4.7601	18

TABLE 3.3: Grain test problem: relative errors (RRE) and CPU times in seconds for standard Tikhonov (2.10), projected Tikhonov (3.3), Algorithm 3.1, and Algorithm 3.4. For the last two methods also the number of iterations is displayed. The smallest error is shown in boldface.



(a) (b) (c) (d)

FIGURE 3.3: Grain test problem reconstructions: (a) standard Tikhonov, (b) projected Tikhonov, (c) Algorithm 3.1, (d) Algorithm 3.4.



(a) (b) (c)

FIGURE 3.4: Peppers test problem: (a) true image (496×496 pixels), (b) motion PSF (21×21 pixels), (c) blurred and noisy image.

Method	RRE	CPU time	Iterations
Tikhonov	0.31718	39.667	–
Projected Tikhonov	0.28645	44.192	–
Algorithm 3.1	0.16160	1.1621×10^3	55
Algorithm 3.4	**0.095639**	33.573	32

TABLE 3.4: Peppers test problem: relative errors (RRE) and CPU times in seconds for standard Tikhonov (2.10), projected Tikhonov (3.3), Algorithm 3.1, and Algorithm 3.4. For the last two methods also the number of iterations is displayed. The smallest error is shown in boldface.

Peppers We now present an example with a larger image. This example illustrates that Algorithm 3.4 may require much less CPU time than Algorithm 3.1. Figure 3.4 displays the true image, the motion PSF used for blurring, and the blurred and noise-contaminated image. The noise is 3% and white Gaussian. Since the image is generic, we impose antireflective boundary conditions.

Similarly as above, we compare standard Tikhonov regularization (2.10), the projected version (3.3), and Algorithms 3.1 and 3.4. We set $\ell = 100$ in the latter algorithm. Table 3.4 provides the relative errors of the computed restorations and the CPU times for the methods. Algorithm 3.4 can be seen to outperform all the other methods both with respect to accuracy in the computed restoration and computing time. In particular, while Algorithm 3.1 yields a restoration of high quality, it requires too much CPU time to be attractive. Figure 3.5 displays the restorations. The imposition of the nonnegativity constraint during the computations can be seen to give a restoration of higher quality than standard and projected Tikhonov regularization (2.10) and (3.3).

(a) (b) (c) (d)

FIGURE 3.5: Peppers test problem reconstructions: (a) standard Tikhonov, (b) projected Tikhonov, (c) Algorithm 3.1, (d) Algorithm 3.4.

Method	RRE	CPU time	Iterations
Tikhonov	0.28354	4.4644	–
Projected Tikhonov	0.26822	4.4766	–
Algorithm 3.1	0.22901	98.349	131
Algorithm 3.5	**0.22757**	1.4647	109

TABLE 3.5: Atmospheric blur test problem: relative errors (RRE) and CPU times in seconds for standard Tikhonov (2.10), projected Tikhonov (3.3), and Algorithm 3.5. For the last method also the number of iterations is displayed.

Atmospheric Blur Our last example considers the test data `AtmoshpericBlur50` from [12]. Figure 3.6 shows the true image, the PSF, and the observed image. Using the knowledge of the true image, we are able to determine an approximation of the noise level in the data, which turns out to be a little more than 1%. Due to the large black area near the boundary, we may impose periodic boundary conditions on the matrix A without significantly reducing the quality of the computed restoration. This makes A a BCCB matrix and matrix-vector products with matrices of the form $A + cI_n$, where c is a scalar, can be computed in only $\mathcal{O}(n\log(n))$ flops with the aid of the FFT. The FFT also can be applied to solve linear systems of equations with a matrix of the form $A+cI_n$ in only $\mathcal{O}(n\log(n))$ flops. We remark that when the available contaminated image is represented by $q \times q$ pixels, each circulant matrix that makes up A is of size $q \times q$, and A has q circulant blocks along the diagonal. Thus, $A \in \mathbb{R}^{n\times n}$ with $n = q^2$.

The spectral factorization (3.25) of A can be computed in $\mathcal{O}(n\log(n))$ flops. This factorization allows the solution of (2.12) for $\alpha > 0$ by Newton's method with each iteration requiring only $\mathcal{O}(n\log(n))$ flops. Using (3.25), we obtain

$$A^tA + \alpha I = F^*(|\Sigma|^2 + \alpha I_n)F.$$

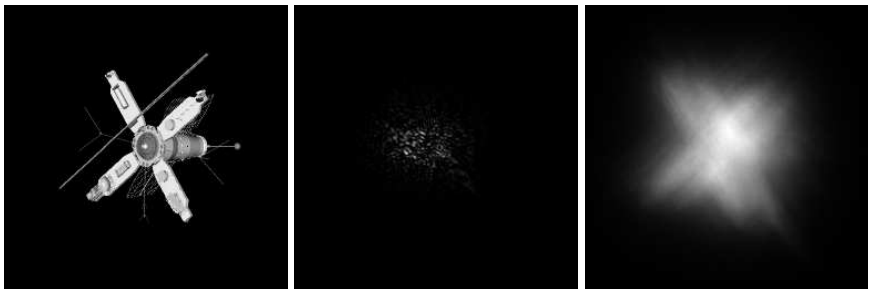
The following algorithm is a modification of Algorithm 3.1 that exploits the BCCB structure of A. It uses the matrix

$$S_\alpha = |\Sigma|^2 + \alpha I_n.$$

Algorithm 3.5 (Fast Fourier Transform Modulus Method)**.** *Compute the decomposition* (3.25) *and determine a regularization parameter* α *that satisfies* (3.21) *as outlined above. Determine the relaxation parameter* (3.24) *and an initial approximate solution* x_0 *of* (3.1).

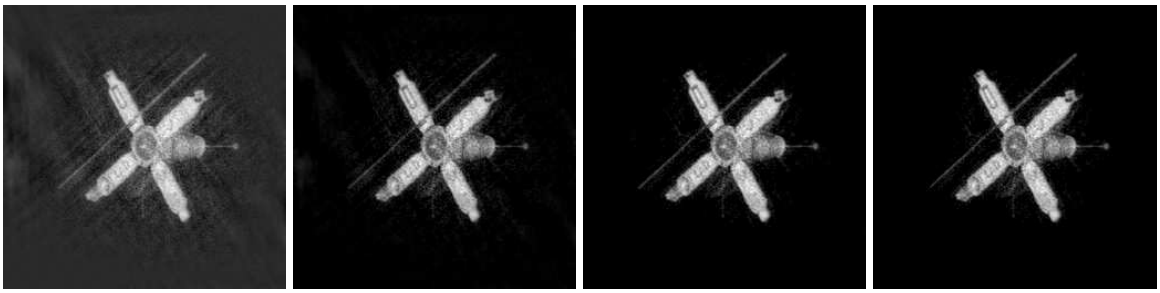
$$\begin{aligned}
&\hat{\mathbf{b}}^\delta = \overline{\Sigma}F\mathbf{b}^\delta\\
&\mathbf{y}_0 = F\mathbf{x}_0\\
&\tilde{\mathbf{y}}_0 = F|F^*\mathbf{y}_0|\\
&\texttt{for } k = 0, 1, 2, \ldots \texttt{until convergence}\\
&\quad \mathbf{y}_{k+1} = (\mu I_n + S_\alpha)^{-1}\left((\mu I_n - S_\alpha)\,\tilde{\mathbf{y}}_k + \hat{\mathbf{b}}^\delta\right)\\
&\quad \tilde{\mathbf{y}}_{k+1} = F|F^*\mathbf{y}_{k+1}|\\
&\texttt{end}\\
&x = F^*\tilde{\mathbf{y}}_{k+1} + |F^*\tilde{\mathbf{y}}_{k+1}|
\end{aligned}$$

Table 3.5 compares the CPU time required and accuracy achieved with Algorithm 3.5 to those for standard and projected Tikhonov regularization (2.10) and (3.3), respectively, and to those for Algorithm 3.1. Algorithm 3.5 imposes periodic boundary conditions, while the other methods are implemented with zero Dirichlet boundary conditions. The table shows Algorithm 3.5 to be the fastest and the one that gives the most accurate restoration. The



(a) (b) (c)

FIGURE 3.6: Atmospheric blur test problem from [12]: (a) true image (256×256 pixels), (b) PSF defined by atmospheric blur (256×256 pixels), (c) blurred and noisy image (256×256 pixels).



(a) (b) (c) (d)

FIGURE 3.7: Atmospheric blur test problem reconstructions: (a) Tikhonov, (b) projected Tikhonov, (c) Algorithm 3.1, (d) Algorithm 3.5.

superior quality of the restoration delivered by Algorithm 3.5 is confirmed by Figure 3.7, which displays the restorations. Algorithm 3.5 can be seen to yield a restoration with a more homogeneous black background than the other methods. Figure 3.8 displays a detail of the lower right corner of the restored images in a different color map.

We do not compare with Algorithm 3.4 in this example, because Algorithm 3.5 yields a more accurate restoration faster than the former. While Algorithm 3.4 performs well for many linear discrete ill-posed problems, Algorithm 3.5 gives superior restorations when the image is such that periodic boundary conditions can be imposed without creating significant boundary artifacts.

Algorithms analogous to Algorithm 3.5 can be developed for reflective and antireflective boundary conditions when the PSF is quadrantally symmetric. For reflective boundary conditions the algorithm can be based on the discrete cosine transform [105] and for antireflective boundary conditions on the antireflective transform (related to the discrete sine transform) [5, 47].

(a) (b) (c) (d)

FIGURE 3.8: Atmospheric blur test problem reconstructions detail (lower right corner) by (a) Tikhonov, (b) projected Tikhonov, (c) Algorithm 3.1 (d) Algorithm 3.5.

Chapter 4

Iterated Tikhonov with general penalty term

It is often possible to improve the quality of the approximation of $\mathbf{x}^\dagger$ determined by Tikhonov regularization by replacing the Tikhonov minimization problem (2.10) by (2.8), i.e., with

$$\min_{\mathbf{x}\in\mathbb{R}^n} \left\{ \left\| A\mathbf{x} - \mathbf{b}^\delta \right\|^2 + \alpha \left\| L(\mathbf{x} - \mathbf{x}_0) \right\|^2 \right\},$$

where $L \in \mathbb{R}^{q\times n}$ is a suitable regularization matrix and $\mathbf{x}_0$ a given approximation of $\mathbf{x}^\dagger$. Recall that, as we saw in (2.9), in order to have a unique solution we need that

$$\mathcal{N}(L) \cap \mathcal{N}(A) = \{\mathbf{0}\}.$$

Like in Section 2.2.2, applying a refinement technique lead to the IT algorithm, cf. Algorithm 2.1. However, available analyses of iterated Tikhonov regularization only treat the case when L is the identity [14, 25, 46, 79]. Computed results reported in [89, 90] show that iterative application of (2.8) with $L \neq I$ can give better approximations of $\mathbf{x}^\dagger$. To the best of our knowledge the only detailed analysis available of iterated Tikhonov regularization

$$\mathbf{x}_{k+1} = \mathbf{x}_k + (A^tA + \alpha_k L^tL)^{-1}A^t(\mathbf{b}^\delta - A\mathbf{x}_k), \qquad k = 0, 1, \ldots, \tag{4.1}$$

with L a fairly general regularization matrix which satisfies (2.9) is the one proposed in [30]. It is the aim of this chapter to illustrate such an analysis and to show that, for suitable choices of L, the iteration (4.1) can give approximations of $\mathbf{x}^\dagger$ of significantly higher quality than the IT iterations. We show that (4.1) defines a regularization method when the iterations are terminated with the discrepancy principle (2.12). Our analysis is first carried out for the stationary IT method with A and L square matrices, and subsequently extended to rectangular matrices and nonstationary iterated Tikhonov regularization.

This chapter is organized as follows: Section 4.1 uses the generalized singular value decomposition of the matrix pair $\{A, L\}$ to derive some results which are needed in the following. The iterated Tikhonov method with a general regularization matrix L is discussed in Section 4.2. We describe the algorithm and discuss properties of the iterates generated. A few computed examples that illustrate the performance of iterated Tikhonov regularization are presented in Section 4.3.

4.1 Standard Tikhonov regularization in general form

Assume that A and L are square matrices, i.e., $m = n = q = d$, and introduce the generalized singular value decomposition (GSVD) of the matrix pair $\{A, L\}$,

$$A = U\Sigma Y^t, \quad L = V\Lambda Y^t, \tag{4.2}$$

where $U, V \in \mathbb{R}^{d\times d}$ are orthogonal matrices, $\Sigma = \mathrm{diag}[\sigma_1, \ldots, \sigma_d] \in \mathbb{R}^{d\times d}$ and $\Lambda = \mathrm{diag}[\lambda_1, \ldots, \lambda_d] \in \mathbb{R}^{d\times d}$ are diagonal matrices, and the matrix $Y \in \mathbb{R}^{d\times d}$ is non-singular. It follows from (2.9) that

$$\sigma_j = 0 \Rightarrow \lambda_j \neq 0 \quad \text{and} \quad \lambda_j = 0 \Rightarrow \sigma_j \neq 0. \tag{4.3}$$

Due to (2.9) the minimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\| A\mathbf{x} - \mathbf{b}^\delta \right\|^2 + \alpha \left\| L\mathbf{x} \right\|^2$$

has the unique solution

$$\mathbf{x}_\alpha = \left(A^tA + \alpha L^tL\right)^{-1} A^t\mathbf{b}^\delta. \tag{4.4}$$

Substituting the factorizations (4.2) into (4.4), we get

$$\begin{aligned}
\mathbf{x}_\alpha &= \left(Y\Sigma U^tU\Sigma Y^t + \alpha Y\Lambda V^tV\Lambda Y^t\right)^{-1} Y\Sigma U^t\mathbf{b}^\delta \\
&= Y^{-t}\left(\Sigma^2 + \alpha\Lambda^2\right)^{-1}\Sigma U^t\mathbf{b} \\
&= Y^{-t}\left(\Sigma^2 + \alpha\Lambda^2\right)^{-1}\Sigma\hat{\mathbf{b}},
\end{aligned}$$

where $\hat{\mathbf{b}} = [\hat{b}_1, \ldots, \hat{b}_d]^t = U^t\mathbf{b}^\delta$. Assume that $\lambda_j = 0$ for $1 \leq j \leq l$, and $\lambda_j \neq 0$ for $l < j \leq d$. Note that, due to (4.3), the ratios $\frac{1}{\sigma_j}$, $1 \leq j \leq l$, are well defined.

$$\begin{aligned}
\mathbf{x}_\alpha &= \sum_{j=1}^{d} \tilde{y}_j \frac{\sigma_j}{\sigma_j^2 + \alpha\lambda_j^2}\hat{b}_j \\
&= \sum_{j=1}^{l} \tilde{y}_j \frac{1}{\sigma_j}\hat{b}_j + \sum_{j=l+1}^{d} \tilde{y}_j \frac{\sigma_j}{\sigma_j^2 + \alpha\lambda_j^2}\hat{b}_j \\
&= \sum_{j=1}^{l} \tilde{y}_j \frac{1}{\sigma_j}\hat{b}_j + \sum_{j=l+1}^{d} \tilde{y}_j \frac{\sigma_j/\lambda_j}{(\sigma_j/\lambda_j)^2 + \alpha}\frac{1}{\lambda_j}\hat{b}_j.
\end{aligned} \tag{4.5}$$

Let us give some definitions that are going to be useful in the following. Introduce the matrix

$$A^{-1}_{\mathcal{N}(L)} = Y^{-t}\begin{pmatrix} 1/\sigma_1 & & & & & & \\ & 1/\sigma_2 & & & & & \\ & & \ddots & & & & \\ & & & 1/\sigma_r & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{pmatrix} U^t = Y^{-t}\Sigma^\dagger(I - \Lambda^\dagger\Lambda)U^t, \tag{4.6}$$

where $r = \min\{l, \operatorname{rank}(A)\}$ and

$$\Lambda^\dagger = \begin{pmatrix} 0 & & & & & \\ & \ddots & & & & \\ & & 0 & & & \\ & & & 1/\lambda_{l+1} & & \\ & & & & \ddots & \\ & & & & & 1/\lambda_d \end{pmatrix}$$

is the pseudo-inverse of Λ. Also define

$$\bar{L} = Y^{-t}\Lambda^\dagger V^t. \tag{4.7}$$

Let $\Gamma = \operatorname{diag}[\gamma_1, \ldots, \gamma_d]$ with $\gamma_j = 0$ for $1 \leq j \leq l$ and $\gamma_j = \frac{\sigma_j}{\lambda_j}$ for $l < j \leq d$. Introduce

$$C = U\Gamma V^t, \tag{4.8}$$

Since the matrices U and V are orthogonal, it follows that (4.8) is the singular value decomposition of C, possibly with the entries of Γ ordered in a non-standard fashion,i.e., it is not assured that $\gamma_j \geq \gamma_{j+1}$ for all γ as in the standard SVD. Combining (4.6)–(4.8) with (4.5), we now can express the solution of (2.8) as

$$\mathbf{x}_\alpha = A^{-1}_{\mathcal{N}(L)}\mathbf{b}^\delta + \bar{L}\left(C^tC + \alpha I\right)^{-1} C^t\mathbf{b}^\delta.$$

4.2 Iterated Tikhonov regularization with a general penalty term

The following algorithm extends iterated Tikhonov regularization with $L = I$ in the stationary case, i.e., with $\alpha_k = \alpha$ for all k, by allowing a fairly general regularization matrix L. The algorithm does not require the matrices A and L to be square.

Algorithm 4.1 (Iterated Tikhonov with general penalty term (GIT)). *Let $A \in \mathbb{R}^{m\times n}$ and $\mathbf{b}^\delta \in \mathbb{R}^m$, and let the regularization matrix $L \in \mathbb{R}^{q\times n}$ satisfy* (2.9)*. Assume that $\delta > 0$ is large enough so that* (2.2) *holds and fix $\tau > 1$ independently of δ. Let $\alpha > 0$ and let $\mathbf{x}_0 \in \mathbb{R}^n$ be an available initial approximation of $\mathbf{x}^\dagger$. Compute*

```
for k = 0, 1, ...
  r_k = b^δ − A x_k
  if ‖r_k‖ < τδ exit
  x_{k+1} = x_k + (A^t A + α L^t L)^{-1} A^t r_k
end
```

In the special case when L is the identity matrix, Algorithm 4.1 simplifies to the IT iterations terminated by the discrepancy principle (2.12). In our analysis of Algorithm 4.1, we first consider the situation when A and L are square matrices. Later, in Subsection 4.2.2, we extend the analysis to more general matrices A and L. Finally, in Subsection 4.2.3, we consider nonstationary sequences of regularization parameters $\alpha_0, \alpha_1, \alpha_2, \ldots$.

4.2.1 Convergence analysis for square matrices A and L

Let $d = m = n = q$. In this subsection we will show that the iterates $\mathbf{x}_k$ determined by Algorithm 4.1 without termination by the discrepancy principle converge to a solution of (2.3). However, as we pointed out in Chapter 2, the solutions of (2.3) are contaminated by propagated error and therefore generally not useful. Typically, a much better approximation of $\mathbf{x}^\dagger$ can be determined by the aid of the discrepancy principle as in Algorithm 4.1. We will show that Algorithm 4.1 is an iterative regularization method.

To show convergence and the regularization property of Algorithm 4.1, we employ a *divide et impera* approach. We set $\mathbf{x}_0 = \mathbf{0}$ in order to simplify the proofs. Consider the iterates

$$\begin{cases} \mathbf{x}_0 = \mathbf{0}, \\ \mathbf{x}_{k+1} = \mathbf{x}_k + (A^tA + \alpha L^tL)^{-1}A^t\mathbf{r}_k, \end{cases}$$

where $\mathbf{r}_k = \mathbf{b}^\delta - A\mathbf{x}_k$ is the residual at step k. Using the expression (4.5), we get that

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + A^{-1}_{\mathcal{N}(L)}\mathbf{r}_k + \bar{L}\left(C^tC + \alpha I\right)^{-1}C^t\mathbf{r}_k \\ &= \sum_{i=0}^{k} A^{-1}_{\mathcal{N}(L)}\mathbf{r}_i + \sum_{i=0}^{k}\bar{L}\left(C^tC + \alpha I\right)^{-1}C^t\mathbf{r}_i. \end{aligned}$$

We will show convergence of the two sums

$$\mathbf{x}^{(0)}_{k+1} = \sum_{i=0}^{k} A^{-1}_{\mathcal{N}(L)}\mathbf{r}_i, \tag{4.9}$$

$$\mathbf{x}^{\perp}_{k+1} = \bar{L}\sum_{i=0}^{k}\left(C^tC + \alpha I\right)^{-1}C^t\mathbf{r}_i, \tag{4.10}$$

for increasing k separately.

Proposition 4.1. *Assume $d = m = n = q$, let $\mathbf{x}^{(0)}_k$ be defined in* (4.9)*, and set $\mathbf{x}_0 = \mathbf{0}$. Then*

$$\mathbf{x}^{(0)}_k = A^{-1}_{\mathcal{N}(L)}\mathbf{b}^\delta \text{ for } k \geq 1.$$

Proof. Since $\mathbf{x}_0 = \mathbf{0}$, we immediately have that

$$\mathbf{x}^{(0)}_1 = A^{-1}_{\mathcal{N}(L)}\mathbf{b}^\delta.$$

It remains to be shown that $\mathbf{x}^{(0)}_k = A^{-1}_{\mathcal{N}(L)}\mathbf{b}^\delta$ for all $k \geq 2$. We proceed by induction. Let $k \geq 1$ and suppose that $\mathbf{x}^{(0)}_k = A^{-1}_{\mathcal{N}(L)}\mathbf{b}^\delta$. Then we need to show that $\mathbf{x}^{(0)}_{k+1} = A^{-1}_{\mathcal{N}(L)}\mathbf{b}^\delta$. We have

$$\begin{aligned} \mathbf{x}^{(0)}_{k+1} &= \mathbf{x}^{(0)}_k + A^{-1}_{\mathcal{N}(L)}(\mathbf{b}^\delta - A\mathbf{x}_k) \\ &= A^{-1}_{\mathcal{N}(L)}\mathbf{b}^\delta + A^{-1}_{\mathcal{N}(L)}(\mathbf{b}^\delta - A(\mathbf{x}^{(0)}_k + \mathbf{x}^{\perp}_k)) \\ &= A^{-1}_{\mathcal{N}(L)}\mathbf{b}^\delta + A^{-1}_{\mathcal{N}(L)}(\mathbf{b}^\delta - AA^{-1}_{\mathcal{N}(L)}\mathbf{b}^\delta - A\mathbf{x}^{\perp}_k). \end{aligned}$$

If we show that $A^{-1}_{\mathcal{N}(L)}(\mathbf{b}^\delta - AA^{-1}_{\mathcal{N}(L)}\mathbf{b}^\delta) = A^{-1}_{\mathcal{N}(L)}A\mathbf{x}_k^\perp = \mathbf{0}$, then the proposition follows. We have that

$$\begin{aligned}
&A^{-1}_{\mathcal{N}(L)}(\mathbf{b}^\delta - AA^{-1}_{\mathcal{N}(L)}\mathbf{b}^\delta) = (A^{-1}_{\mathcal{N}(L)} - A^{-1}_{\mathcal{N}(L)}AA^{-1}_{\mathcal{N}(L)})\mathbf{b}^\delta \\
&= (Y^{-t}\Sigma^\dagger(I-\Lambda^\dagger\Lambda)U^t - Y^{-t}\Sigma^\dagger(I-\Lambda^\dagger\Lambda)U^tU\Sigma Y^tY^{-t}\Sigma^\dagger(I-\Lambda^\dagger\Lambda)U^t)\mathbf{b}^\delta \\
&= Y^{-t}(\Sigma^\dagger(I-\Lambda^\dagger\Lambda) - \Sigma^\dagger(I-\Lambda^\dagger\Lambda)\Sigma\Sigma^\dagger(I-\Lambda^\dagger\Lambda))U^t\mathbf{b}^\delta \\
&= Y^{-t}(\Sigma^\dagger(I-\Lambda^\dagger\Lambda) - \Sigma^\dagger\Sigma\Sigma^\dagger(I-\Lambda^\dagger\Lambda)(I-\Lambda^\dagger\Lambda))U^t\mathbf{b}^\delta \\
&= Y^{-t}(\Sigma^\dagger(I-\Lambda^\dagger\Lambda) - \Sigma^\dagger(I-\Lambda^\dagger\Lambda))U^t\mathbf{b}^\delta \\
&= \mathbf{0},
\end{aligned}$$

where we have used the facts that diagonal matrices commute, that $\Sigma^\dagger\Sigma\Sigma^\dagger = \Sigma^\dagger$, and that $(I-\Lambda^\dagger\Lambda)(I-\Lambda^\dagger\Lambda) = (I-\Lambda^\dagger\Lambda)$, since $(I-\Lambda^\dagger\Lambda)$ is an orthogonal projector.

Turning to $A^{-1}_{\mathcal{N}(L)}A\mathbf{x}_k^\perp$, we prove that $A^{-1}_{\mathcal{N}(L)}A\bar{L} = 0$. We get

$$\begin{aligned}
A^{-1}_{\mathcal{N}(L)}A\bar{L} &= Y^{-t}\Sigma^\dagger(I-\Lambda^\dagger\Lambda)U^tU\Sigma Y^tY^{-t}\Lambda^\dagger V^t \\
&= Y^{-t}\Sigma^\dagger\Sigma(I-\Lambda^\dagger\Lambda)\Lambda^\dagger V^t \\
&= Y^{-t}\Sigma^\dagger\Sigma(\Lambda^\dagger - \Lambda^\dagger\Lambda\Lambda^\dagger)V^t \\
&= Y^{-t}\Sigma^\dagger\Sigma(\Lambda^\dagger - \Lambda^\dagger)V^t \\
&= 0.
\end{aligned}$$

It follows that $A^{-1}_{\mathcal{N}(L)}A\mathbf{x}_k^\perp = \mathbf{0}$ by induction because

$$A^{-1}_{\mathcal{N}(L)}A\mathbf{x}_k^\perp = A^{-1}_{\mathcal{N}(L)}A\mathbf{x}_{k-1}^\perp + A^{-1}_{\mathcal{N}(L)}A\bar{L}(C^tC+\alpha I)^{-1}C^t(b - A\mathbf{x}_k),$$

which concludes the proof. □

Proposition 4.2. *Let $d = m = n = q$ and assume that* (2.9) *holds. Let $\mathbf{x}_k^\delta$ be defined in* (4.10) *and set $\mathbf{x}_0 = \mathbf{0}$. Then*

$$\mathbf{x}_k^\perp \to \bar{L}C^\dagger\bar{\mathbf{b}}^\delta \text{ as } k \to \infty,$$

where

$$\bar{\mathbf{b}}^\delta = U\Lambda^\dagger\Lambda U^t\mathbf{b}^\delta.$$

Proof. Consider the sequence $\{\mathbf{x}_k^\perp\}_{k=1}^\infty$. We would like to show that this sequence can be determined by application of standard iterated Tikhonov regularization to some linear system of equations. The convergence then will follow from available results for iterative Tikhonov regularization with regularization matrix $L = I$. First recall the expression for $\mathbf{x}_{k+1}^\perp$,

$$\mathbf{x}_{k+1}^\perp = \mathbf{x}_k^\perp + \bar{L}(C^tC+\alpha I)^{-1}C^t(\mathbf{b}^\delta - A\mathbf{x}_k).$$

To transform this iteration to (standard) iterated Tikhonov iterations, we introduce

$$\tilde{\mathbf{h}}_k = (C^tC+\alpha I)^{-1}C^t(\mathbf{b}^\delta - A\mathbf{x}_k), \tag{4.11}$$

such that

$$\mathbf{x}_{k+1}^\perp = \mathbf{x}_k^\perp + \bar{L}\tilde{\mathbf{h}}_k. \tag{4.12}$$

Inserting the factorizations (4.2) and (4.8) of A and C into (4.11) yields

$$\tilde{\mathbf{h}}_k = V(\Gamma^2+\alpha I)^{-1}\Gamma U^t(\mathbf{b}^\delta - U\Sigma Y^t\mathbf{x}_k) = V(\Gamma^2+\alpha I)^{-1}\Gamma(U^t\mathbf{b}^\delta - \Sigma Y^t\mathbf{x}_k).$$

We have

$$\Gamma\Sigma = \Gamma\Gamma\Lambda,$$

because both the left-hand and right-hand sides are diagonal matrices whose first l components vanish, and the remaining components are of the form σ_j^2/λ_j for $l < j \leq d$. Thus, we obtain

$$\tilde{\mathbf{h}}_k = V(\Gamma^2 + \alpha I)^{-1}\Gamma(U^t b - \Gamma\Lambda Y^t \mathbf{x}_k).$$

Define

$$\bar{\mathbf{b}}^\delta = U\Lambda^\dagger\Lambda U^t \mathbf{b}^\delta$$

and

$$\bar{\mathbf{x}}_k = L\mathbf{x}_k,$$

and consider

$$\bar{\mathbf{h}}_k = (C^tC + \alpha I)^{-1}C^t(\bar{\mathbf{b}}^\delta - C\bar{\mathbf{x}}_k).$$

We will show that $\bar{\mathbf{h}}_k = \tilde{\mathbf{h}}_k$. Substituting the factorizations (4.8) and (4.2) of C and L into the above expression, we get

$$\begin{aligned}
\bar{\mathbf{h}}_k &= V(\Gamma^2 + \alpha I)^{-1}V^tV\Gamma U^t(U\Lambda^\dagger\Lambda U^t\mathbf{b}^\delta - U\Gamma V^tV\Lambda Y^t\mathbf{x}_k) \\
&= V(\Gamma^2 + \alpha I)^{-1}\Gamma(U^t\mathbf{b}^\delta - \Gamma\Lambda Y^t\mathbf{x}_k) \\
&= \tilde{\mathbf{h}}_k,
\end{aligned}$$

where in the last step we have used the fact that $\Gamma\Lambda^\dagger\Lambda = \Gamma$. Replacing $\tilde{\mathbf{h}}_k$ by $\bar{\mathbf{h}}_k$ in (4.12), we obtain

$$\mathbf{x}_{k+1}^\perp = \mathbf{x}_k^\perp + \bar{L}\bar{\mathbf{h}}_k = \mathbf{x}_k^\perp + \bar{L}(C^tC + \alpha I)^{-1}C^t(\bar{\mathbf{b}}^\delta - CL\mathbf{x}_k).$$

Since $\mathbf{x}_0 = \mathbf{0}$, we have

$$\mathbf{x}_{k+1}^\perp = \bar{L}\sum_{i=0}^{k}(C^tC + \alpha I)^{-1}C^t(\bar{\mathbf{b}}^\delta - CL\mathbf{x}_i).$$

We now show that the sum in the right-hand side, namely

$$\tilde{\mathbf{x}}_{k+1} = \sum_{i=0}^{k}(C^tC + \alpha I)^{-1}C^t(\bar{\mathbf{b}}^\delta - CL\mathbf{x}_i)$$

is the approximate solution computed by $k+1$ iterations of standard iterated Tikhonov iteration applied to the linear system of equations

$$C\mathbf{x} = \bar{\mathbf{b}}^\delta. \tag{4.13}$$

We have

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{x}}_k + (C^tC + \alpha I)^{-1}C^t(\bar{\mathbf{b}}^\delta - CL\mathbf{x}_k).$$

Therefore, if we establish that $L\mathbf{x}_k = \tilde{\mathbf{x}}_k$ for all k, then we are done. We show this result by induction. For $k = 0$ it is trivial since $\mathbf{x}_0 = \mathbf{0}$. Suppose that $\tilde{\mathbf{x}}_k = L\mathbf{x}_k$. We would like to show

that $\tilde{\mathbf{x}}_{k+1} = L\mathbf{x}_{k+1}$. Applying L to $\mathbf{x}_{k+1}$ yields

$$\begin{aligned}
L\mathbf{x}_{k+1} &= L\mathbf{x}_k + LA^{-1}_{\mathcal{N}(L)}\mathbf{r}_k + L\bar{L}(C^tC+\alpha I)^{-1}C^t(\mathbf{b}^\delta - A\mathbf{x}_k)\\
&\overset{(a)}{=} \tilde{\mathbf{x}}_k + \mathbf{0} + L\bar{L}(C^tC+\alpha I)^{-1}C^t(\mathbf{b}^\delta - A\mathbf{x}_k)\\
&\overset{(b)}{=} \tilde{\mathbf{x}}_k + L\bar{L}(C^tC+\alpha I)^{-1}C^t(\bar{\mathbf{b}}^\delta - CL\mathbf{x}_k)\\
&= \tilde{\mathbf{x}}_k + V\Lambda^\dagger Y^t Y^{-t}\Lambda V^t V(\Gamma^2+\alpha I)^{-1}\Gamma U^t(\bar{\mathbf{b}}^\delta - CL\mathbf{x}_k)\\
&= \tilde{\mathbf{x}}_k + V\Lambda^\dagger\Lambda(\Gamma^2+\alpha I)^{-1}\Gamma U^t(\bar{\mathbf{b}}^\delta - CL\mathbf{x}_k)\\
&\overset{(c)}{=} \tilde{\mathbf{x}}_k + V(\Gamma^2+\alpha I)^{-1}\Gamma U^t(\bar{\mathbf{b}}^\delta - CL\mathbf{x}_k)\\
&= \tilde{\mathbf{x}}_k + (C^tC+\alpha I)^{-1}C^t(\bar{\mathbf{b}}^\delta - CL\mathbf{x}_k)\\
&= \tilde{\mathbf{x}}_{k+1},
\end{aligned}$$

where equality (a) is due to the fact that $A^{-1}_{\mathcal{N}(L)}$ annihilates the component of $\mathbf{r}_k = \mathbf{b}^\delta - A\mathbf{x}_k$ in the complement of $\mathcal{N}(L)$, (b) is obtained by using the fact, shown above, that $\tilde{\mathbf{h}}_k = \bar{\mathbf{h}}_k$, and (c) follows from $\Lambda^\dagger\Lambda\Gamma = \Gamma$.

We have shown that the $\tilde{\mathbf{x}}_k$ are iterates determined by the (standard) iterated Tikhonov method applied to the linear system of equations (4.13) and thus it follows that

$$\tilde{\mathbf{x}}_k \to C^\dagger\bar{\mathbf{b}}^\delta \text{ as } k\to\infty,$$

due to the convergence of the iterated Tikhonov method [61]. By continuity of $\bar{L}$, we have that

$$\mathbf{x}_k^\perp \to \bar{L}C^\dagger\bar{\mathbf{b}}^\delta \text{ as } k\to\infty,$$

which concludes the proof. □

Introduce the matrix

$$A^{(\dagger)} = Y^{-t}\Sigma^\dagger U^t.$$

Theorem 4.3 (Convergence). *Let $d = m = n = q$ and assume that (2.9) holds. Let $\mathbf{x}_0 = \mathbf{0}$. Then the iterates determined by Algorithm 4.1 converge to $A^{(\dagger)}\mathbf{b}^\delta$. Moreover, if $\mathbf{b}^\delta \in \mathcal{R}(A)$, then $AA^{(\dagger)}\mathbf{b}^\delta = \mathbf{b}^\delta$.*

Proof. From Propositions 4.1 and 4.2, we have

$$\mathbf{x}_k = \mathbf{x}_k^{(0)} + \mathbf{x}_k^\perp \to A^{-1}_{\mathcal{N}(L)}\mathbf{b}^\delta + \bar{L}C^\dagger\bar{\mathbf{b}}^\delta = \mathbf{x}_\infty \text{ as } k\to\infty.$$

Using the definitions (4.6), (4.7), and (4.8), we obtain

$$\begin{aligned}
\mathbf{x}_\infty &= Y^{-t}\Sigma^\dagger(I-\Lambda^\dagger\Lambda)U^t\mathbf{b}^\delta + Y^{-t}\Lambda^\dagger V^t V\Gamma^\dagger U^t U\Lambda^\dagger\Lambda U^t\mathbf{b}^\delta\\
&= Y^{-t}\left(\Sigma^\dagger(I-\Lambda^\dagger\Lambda) + \Lambda^\dagger\Gamma^\dagger\Lambda^\dagger\Lambda\right)U^t\mathbf{b}^\delta\\
&= Y^{-t}\left(\Sigma^\dagger(I-\Lambda^\dagger\Lambda) + \Lambda^\dagger\Gamma^\dagger\right)U^t\mathbf{b}^\delta\\
&= Y^{-t}\left(\Sigma^\dagger(I-\Lambda^\dagger\Lambda) + \Lambda^\dagger\Lambda\Sigma^\dagger\right)U^t\mathbf{b}^\delta\\
&= Y^{-t}\Sigma^\dagger U^t\mathbf{b}^\delta,
\end{aligned}$$

where we have used the fact that diagonal matrices commute and $\Lambda^\dagger\Gamma^\dagger = \Lambda^\dagger\Lambda\Sigma^\dagger$.

What is left to prove is that if $\mathbf{b}^\delta \in \mathcal{R}(A)$, then $AA^{(\dagger)}\mathbf{b}^\delta = \mathbf{b}^\delta$, which is straightforward. Since $\mathbf{b}^\delta \in \mathcal{R}(A)$, there exists $\mathbf{y} \in \mathbb{R}^d$ such that $\mathbf{b}^\delta = A\mathbf{y}$ thus

$$\begin{aligned} AA^{(\dagger)}\mathbf{b}^\delta &= AA^{(\dagger)}A\mathbf{y} \\ &= U\Sigma Y^t Y^{-t}\Sigma^\dagger U^t U\Sigma Y^t \mathbf{y} \\ &= U\Sigma\Sigma^\dagger \Sigma Y^t \mathbf{y} \\ &= U\Sigma Y^t \mathbf{y} = A\mathbf{y} = \mathbf{b}^\delta, \end{aligned}$$

which concludes the proof. □

Remark 4.4. *We note that $\mathbf{x}_\infty = A^{(\dagger)}\mathbf{b}^\delta$ might not be the minimum norm solution of the system* (2.3)*, because $\mathbf{x}_\infty$ may have a component in $\mathcal{N}(A)$.*

Theorem 4.3 shows that the iterates determined by Algorithm 4.1 converge to a solution of (2.3), when A is a square matrix, for any fixed regularization parameter $\alpha > 0$. This result is useful when the vector $\mathbf{b}^\delta$ is error-free, i.e., when $\delta = 0$ in (2.2). However, as already mentioned, when $\mathbf{b}^\delta$ is error-contaminated, the minimum norm solution $A^\dagger\mathbf{b}^\delta$ typically is severely contaminated by propagated error stemming from the error in $\mathbf{b}^\delta$ and, therefore, is not useful. Moreover, the solution $A^{(\dagger)}\mathbf{b}^\delta$ typically is not useful either. A meaningful approximation of $\mathbf{x}^\dagger$ can be determined by terminating the iterations sufficiently early. We will show that the discrepancy principle can be applied to determine when to terminate the iterations. This requires the following auxiliary result.

Lemma 4.5. *Assume that $d = m = n = q$ and that* (2.9) *holds. Let $\delta > 0$, $\mathbf{b} \in \mathcal{R}(A)$, and $\mathbf{x}_0 = \mathbf{0}$. Then Algorithm 4.1 terminates after finitely many steps.*

Proof. Consider the residual at the limit point

$$\mathbf{r}_k \to \mathbf{r}_\infty = \mathbf{b}^\delta - AA^{(\dagger)}\mathbf{b}^\delta = \left(I - AA^{(\dagger)}\right)(\mathbf{b} + \boldsymbol{\eta}) = \left(I - AA^{(\dagger)}\right)\boldsymbol{\eta},$$

where we have called $\boldsymbol{\eta} = \mathbf{b}^\delta - \mathbf{b}$ and in the last step we have used that $\mathbf{b} \in \mathcal{R}(A)$. Now, by (2.2), we have

$$\|\mathbf{r}_\infty\| = \left\|\left(I - AA^{(\dagger)}\right)\boldsymbol{\eta}\right\| \overset{(a)}{\leq} \|\boldsymbol{\eta}\| \leq \delta,$$

where the inequality (a) follows from the fact that $I - AA^{(\dagger)}$ is an orthogonal projector; we have

$$I - AA^{(\dagger)} = I - U\Sigma Y^t Y^{-t}\Sigma^\dagger U^t = U(I - \Sigma\Sigma^\dagger)U^t,$$

where U is an orthogonal matrix.

Let $\tau > 1$ be a constant independent of δ. Then there is a constant $k_\tau < \infty$ such that for all $k > k_\tau$, it holds

$$\|\mathbf{r}_k\| < \tau\delta.$$

□

We are now able to prove the regularization property of Algorithm 4.1.

Theorem 4.6 (Regularization)**.** *Let $\mathbf{b} \in \mathcal{R}(A)$. Then, under the assumptions of Theorem 4.3 and Lemma 4.5, Algorithm 4.1 terminates as soon as a residual vector $\mathbf{r}_k = \mathbf{b}^\delta - A\mathbf{x}_k$ satisfies $\|\mathbf{r}_k\| \leq \tau\delta$. This stopping criterion is satisfied after finitely many steps $k = k_\delta$. Denote the iterate $\mathbf{x}_{k_\delta}$ simply by $\mathbf{x}^\delta$. Then*

$$\limsup_{\delta \searrow 0} \left\|\mathbf{x}^{(\dagger)} - \mathbf{x}^\delta\right\| = 0,$$

where $\mathbf{x}^{(\dagger)} = A^{(\dagger)}\mathbf{b}$.

Proof. It follows from Lemma 4.5 that if $\delta > 0$, then the iterations with Algorithm 4.1 are terminated after finitely many, k, steps. Since $\mathbf{x}_0 = \mathbf{0}$, the iterates determined by the algorithm can be expressed as

$$\mathbf{x}_k = \sum_{j=0}^{k-1} \mathbf{h}_j,$$

where

$$\mathbf{h}_j = A_{\mathcal{N}(L)}^{-1}\mathbf{r}_j + \bar{L}(C^tC + \alpha I)^{-1}C^t\mathbf{r}_j.$$

We first show that

$$A^{(\dagger)}A\mathbf{x}^\delta = \mathbf{x}^\delta.$$

Consider

$$\begin{aligned}
A^{(\dagger)}A\mathbf{h}_j &= A^{(\dagger)}A(A_{\mathcal{N}(L)}^{-1} + \bar{L}(C^tC + \alpha I)^{-1}C^t)\mathbf{r}_j \\
&= Y^{-t}\Sigma^\dagger\Sigma Y^t(Y^{-t}\Sigma^\dagger(I - \Lambda^\dagger\Lambda)U^t + Y^{-t}\Lambda^\dagger(\Gamma^2 + \alpha I)^{-1}\Gamma^t U^t)\mathbf{r}_j \\
&= (Y^{-t}\Sigma^\dagger\Sigma\Sigma^\dagger(I - \Lambda^\dagger\Lambda)U^t + Y^{-t}\Sigma^\dagger\Sigma\Lambda^\dagger(\Gamma^2 + \alpha I)^{-1}\Gamma^t U^t)\mathbf{r}_j \\
&= (Y^{-t}\Sigma^\dagger(I - \Lambda^\dagger\Lambda)U^t + Y^{-t}\Lambda^\dagger(\Gamma^2 + \alpha I)^{-1}\Gamma^t U^t)\mathbf{r}_j \\
&= \mathbf{h}_j.
\end{aligned}$$

Thus, we obtain

$$A^{(\dagger)}A\mathbf{x}^\delta = \sum_{j=0}^{k_\delta-1} A^{(\dagger)}A\mathbf{h}_j = \sum_{j=0}^{k_\delta-1} \mathbf{h}_j = \mathbf{x}^\delta.$$

Therefore,

$$\begin{aligned}
\limsup_{\delta\searrow 0} \left\|\mathbf{x}^{(\dagger)} - \mathbf{x}^\delta\right\| &= \limsup_{\delta\searrow 0} \left\|A^{(\dagger)}A\left(\mathbf{x}^{(\dagger)} - \mathbf{x}^\delta\right)\right\| \\
&\le \left\|A^{(\dagger)}\right\| \limsup_{\delta\searrow 0} \left\|A\left(\mathbf{x}^{(\dagger)} - \mathbf{x}^\delta\right)\right\| \\
&= \left\|A^{(\dagger)}\right\| \limsup_{\delta\searrow 0} \left\|\left(\mathbf{b} - \mathbf{b}^\delta\right) + \left(\mathbf{b}^\delta - A\mathbf{x}^\delta\right)\right\| \\
&\le \left\|A^{(\dagger)}\right\| \limsup_{\delta\searrow 0} (1 + \tau)\,\delta \\
&= 0,
\end{aligned}$$

where in the last step we have used that $\mathbf{x}^\delta$ is determined by the discrepancy principle. □

Remark 4.7. *As already mentioned,* $A^{(\dagger)}\mathbf{b}$ *might not be a minimum norm solution with respect to the Euclidean vector norm. Instead, it is a minimum norm solution with respect to a vector norm induced by the matrix* Y*. We have*

$$\left\|A^{(\dagger)}\mathbf{b}\right\| = \left\|Y^{-t}\Sigma U^t\mathbf{b}\right\| = \left\|\Sigma U^t\mathbf{b}\right\|_{Y^{-t}},$$

where we define the norm induced by an invertible matrix $M \in \mathbb{R}^{d\times d}$ *as* $\|\mathbf{y}\|_M = \|M\mathbf{y}\|$*; see, e.g, [88, eq. (5.6.2)]. The norm in the right-hand side is determined by* Y^{-1}*, which, in turn, is defined by the GSVD of the matrix pair* $\{A, L\}$*.*

4.2.2 Extension of the convergence analysis to rectangular matrices A and L

We show how the analysis of the previous subsection for square matrices A and L can be extended to rectangular matrices. First consider the case when $A \in \mathbb{R}^{m\times n}$ with $m < n$. We then pad A and $\mathbf{b}^\delta$ with $n - m$ zero rows to obtain

$$\widehat{A} = \begin{bmatrix} A \\ O \end{bmatrix} \in \mathbb{R}^{n\times n}, \qquad \widehat{\mathbf{b}}^\delta = \begin{bmatrix} \mathbf{b}^\delta \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^n,$$

and replace A and $\mathbf{b}^\delta$ in (2.3) by $\widehat{A}$ and $\widehat{\mathbf{b}}^\delta$, respectively. This replacement does not change the solution of the minimization problem (2.3).

The situation when $A \in \mathbb{R}^{m\times n}$ with $m > n$ can be handled by padding A with $m - n$ zero columns and the solution $\mathbf{x}$ with $m - n$ zero rows. We obtain

$$\widehat{A} = [A\ 0] \in \mathbb{R}^{m\times m}, \qquad \widehat{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^m,$$

and replace A and $\mathbf{x}$ in (2.3) by $\widehat{A}$ and $\widehat{\mathbf{x}}$. Only the n first entries of the computed solution are of interest.

The case when $L \in \mathbb{R}^{q\times n}$ with $q < n$ can be treated similarly as when A has fewer rows than columns. Thus, we pad L with $n - q$ zero rows to obtain

$$\widehat{L} = \begin{bmatrix} L \\ O \end{bmatrix} \in \mathbb{R}^{n\times n},$$

and replace L in (2.8) by $\widehat{L}$. This replacement does not affect the computed solution.

Finally, when $L \in \mathbb{R}^{q\times n}$ with $q > n$, we compute the QR factorization

$$L = QR,$$

where $Q \in \mathbb{R}^{q\times n}$ has orthonormal columns and $R \in \mathbb{R}^{n\times n}$ is upper triangular. We then replace L in (2.8) by R. The computed solution is not affected by this replacement.

4.2.3 The nonstationary iterated Tikhonov method with a general L

This section extends the analysis of the stationary iterated Tikhonov regularization method described in Subsection 4.2.1 and implemented by Algorithm 4.1 to nonstationary iterated Tikhonov regularization. This extension can be carried out in a fairly straightforward manner. We therefore only state the results and give sketches of proofs.

Consider the iterations

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \left(A^tA + \alpha_k L^tL\right)^{-1} A^t\mathbf{r}_k, \quad k = 0, 1, \dots\ ,$$

where as usual $\mathbf{r}_k$ denotes the residual vector. We assume that (2.9) holds and that the regularization parameters $\alpha_k > 0$ satisfy

$$\sum_{k=0}^{\infty} \alpha_k^{-1} = \infty. \tag{4.14}$$

Analyses of this iteration method when $L = I$ are presented in [25, 79]. The following algorithm outlines the computations with the discrepancy principle as stopping criterion.

Algorithm 4.2 (GIT$_{NS}$). *Let $A \in \mathbb{R}^{m\times n}$, $\mathbf{b}^\delta \in \mathbb{R}^m$, and $\mathbf{x} \in \mathbb{R}^n$. Assume that the regularization matrix $L \in \mathbb{R}^{q\times n}$ satisfies* (2.9) *and that the regularization parameters $\alpha_k > 0$ satisfy* (4.14). *Let δ be defined in* (2.2) *and fix $\tau > 1$ independently of δ. Let $\mathbf{x}_0 \in \mathbb{R}^n$ be an available initial approximation of $\mathbf{x}^\dagger$. Compute*

```
for k = 0, 1, ...
  r_k = b^δ − A x_k
  if ‖r_k‖ < τδ exit.
  x_{k+1} = x_k + (A^t A + α_k L^t L)^{-1} A^t r_k
end.
```

We would like to show that, under the assumption (4.14), the iterates determined by the above algorithm without the stopping criterion converge to $A^{(\dagger)}\mathbf{b}^\delta$, and that the algorithm with stopping criterion defines a regularization method. In the remainder of this section, we only consider square matrices A and L. Extensions to rectangular matrices follow as described in Subsection 4.2.2.

Theorem 4.8 (Convergence). *Assume that $m = n = q$ and that* (2.9) *holds. Let the regularization parameters $\alpha_k > 0$ satisfy* (4.14). *Then the iterates determined by Algorithm 4.2 without stopping criterion converge to the solution $A^{(\dagger)}\mathbf{b}^\delta$ of the linear system of equations $A\mathbf{x} = \mathbf{b}^\delta$.*

Proof. The result can be shown in a similar fashion as Theorem 4.3. We therefore only outline the proof. Similarly as in the proof of Propositions 4.1 and 4.2, we split the iterates as

$$\mathbf{x}_k = \mathbf{x}_k^{(0)} + \mathbf{x}_k^{\perp}.$$

Using the GSVD (4.2) we can show that

$$\mathbf{x}_k^{(0)} \to A^{-1}_{\mathcal{N}(L)}\mathbf{b}^\delta \text{ as } k \to \infty, \tag{4.15}$$

$$\mathbf{x}_k^{\perp} \to \bar{L}C^\dagger\bar{\mathbf{b}}^\delta \text{ as } k \to \infty. \tag{4.16}$$

Similarly as in Proposition 4.1, one can show that $\mathbf{x}_k^{(0)} = A^{-1}_{\mathcal{N}(L)}\mathbf{b}^\delta$ for all k. For the $\mathbf{x}_k^{\perp}$ it holds that

$$\mathbf{x}_{k+1}^{\perp} = \bar{L}\tilde{\mathbf{x}}_{k+1} = \sum_{i=0}^{k}(C^tC + \alpha_i I)^{-1}C^t(\bar{\mathbf{b}}^\delta - C\tilde{\mathbf{x}}_i).$$

Using the assumption (4.14) and [25, Theorem 1.4 p. 21], it follows that

$$\tilde{\mathbf{x}}_k \to C^\dagger\bar{\mathbf{b}}^\delta \text{ as } k \to \infty.$$

By continuity of $\bar{L}$, we obtain

$$\mathbf{x}_k^{\perp} \to \bar{L}C^\dagger\bar{\mathbf{b}}^\delta \text{ as } k \to \infty.$$

Combining (4.15) and (4.16) shows the theorem. □

The following result follows similarly as Theorem 4.6. We therefore omit the proof.

Theorem 4.9 (Regularization). *Let the assumptions of Theorem 4.8 and Lemma 4.5 hold. Then Algorithm 4.2 (with stopping criterion) terminates when a residual vector $\mathbf{r}_k = \mathbf{b}^\delta - A\mathbf{x}_k$ satisfies $\|\mathbf{r}_k\| \le \tau\delta$. This stopping criterion is satisfied after finitely many steps $k = k_\delta$. Denote the iterate*

$\mathbf{x}_{k_\delta}$ *simply by* $\mathbf{x}^\delta$. *Then*

$$\limsup_{\delta \searrow 0} \left\| \mathbf{x}^{(\dagger)} - \mathbf{x}^\delta \right\| = 0.$$

4.3 Numerical examples

This section presents some computed examples where we illustrate the performances of both stationary and nonstationary iterated Tikhonov method with general penalty term, referred to as GIT and GIT$_{NS}$, respectively. We first consider three test problems in one space-dimension. These problems are taken from REGULARIZATION TOOLS [83]. Subsequently an image restoration example in two-dimensional space is considered.

The $n \times n$ bidiagonal and tridiagonal matrices

$$L_1 = \begin{pmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ & & & 0 \end{pmatrix}, \qquad L_2 = \begin{pmatrix} 0 & 0 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & 0 & 0 \end{pmatrix}.$$

are scaled discretizations of the first and second derivative operators at equidistant points in one space-dimension. Their null spaces are

$$L_1 = \operatorname{span}\left\{ \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \right\}, \qquad \mathrm{L}_2 = \operatorname{span}\left\{ \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \\ \vdots \\ n \end{pmatrix} \right\}.$$

The matrix L_1 preserves sampling of constant functions, while L_2 also preserves uniform sampling of linear functions; see [53].

We apply the GIT$_{NS}$ algorithm using the geometric sequence of regularization parameters (2.20). They satisfy

$$\sum_{k=0}^{\infty} \alpha_k^{-1} = \frac{1}{\alpha_0} \sum_{k=0}^{\infty} \frac{1}{q^k} = \infty,$$

which shows that the hypothesis on the regularization parameters of Theorems 4.8 and 4.9 hold. We fix $q = 0.8$, while the choice of α_0 will depend on L. The relative reconstruction error of the computed solution $\mathbf{x}_k$ is measured by means of the RRE.

We compare the GIT and GIT$_{NS}$ methods to classical iterated Tikhonov methods with stationary and non-stationary sequences of regularization parameters, referred to as IT and IT$_{NS}$, respectively. We recall that IT and IT$_{NS}$ can be obtained as special cases of GIT and GIT$_{NS}$, respectively, by choosing $L = I$. All problems in one space-dimension have square matrices $A \in \mathbb{R}^{1000 \times 1000}$. The matrices A and error-free vectors $\mathbf{b}$ are determined by MATLAB functions in [83]. We define the error-contaminated vector $\mathbf{b}^\delta$ by adding white Gaussian noise to $\mathbf{b}$ with a user-chosen noise level ξ.

The iterations with all methods in our comparison are terminated with the discrepancy principle (2.19) with $\tau = 1.01$.

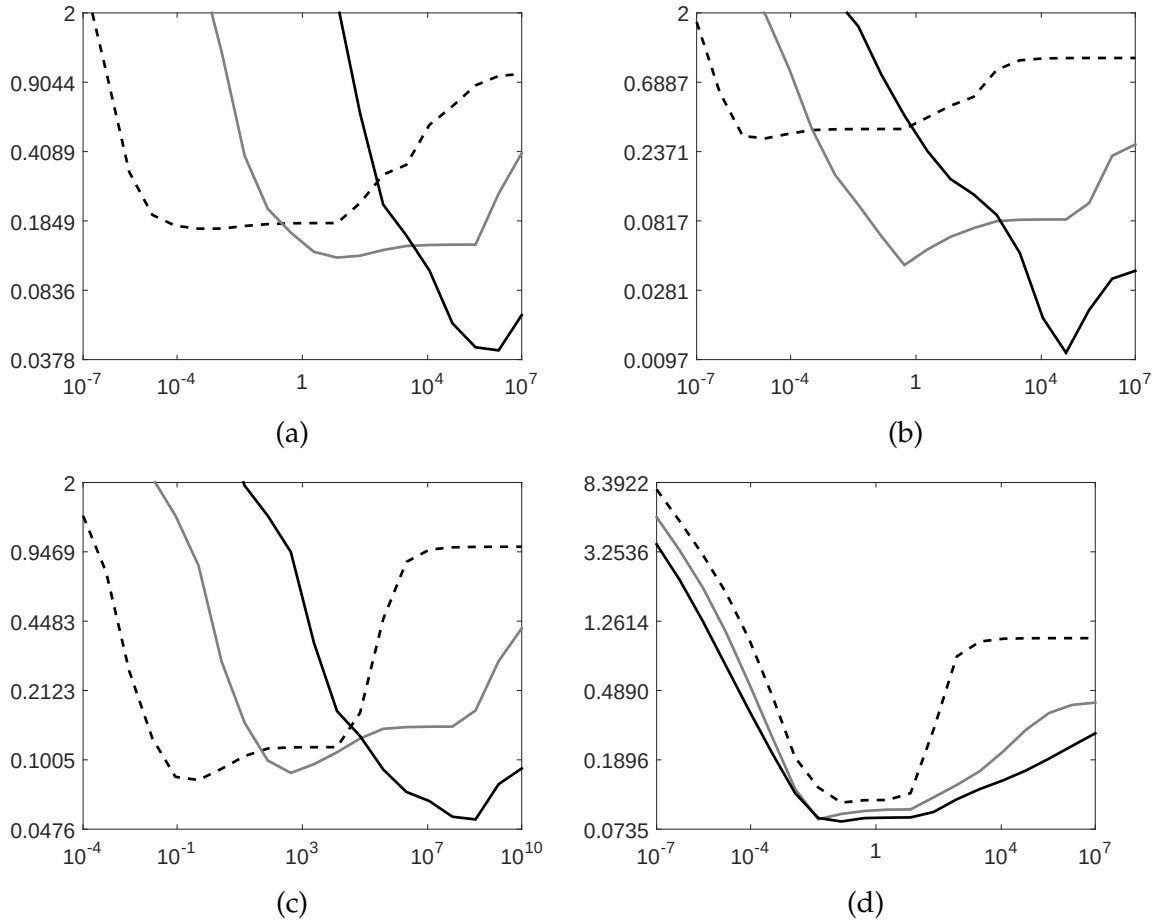


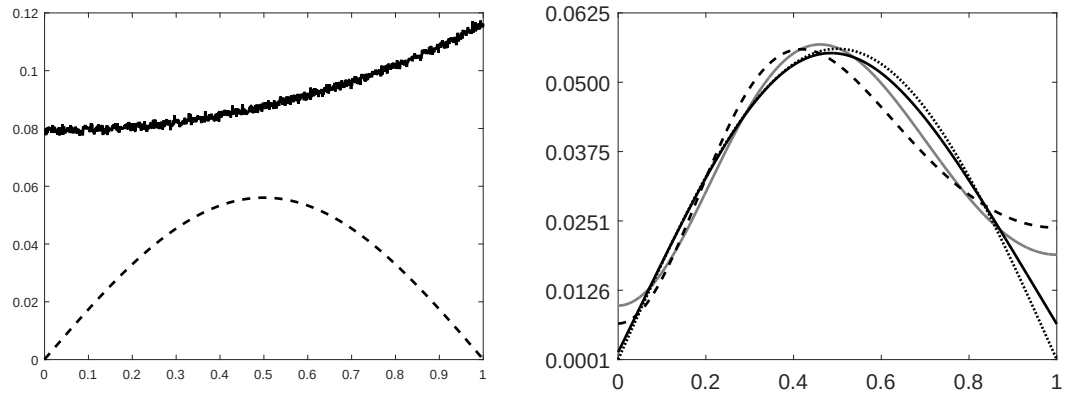
FIGURE 4.1: Stationary iterated Tikhonov regularization: RRE for the iterate determined by the discrepancy principle for different values of α. (a) Baart test problem, (b) Deriv2 test problem, (c) Gravity test problem, (d) Peppers test problem. The dashed curves are for $L = I$, the solid gray curves for $L = L_1$, and the solid black curves for $L = L_2$.

As stated in Remark 4.7, the computed solution may have a component in $\mathcal{N}(A)$. The size of this component depends on the matrix Y in (4.2). We will tabulate the norm of this component for the examples in one space-dimension. The orthogonal projector $P_{\mathcal{N}(A)}$ onto $\mathcal{N}(A)$ is computed with the aid of the SVD of A. We set all singular values smaller than machine epsilon to zero and compute

$$\frac{\left\|P_{\mathcal{N}(A)}\mathbf{x}^{\delta}\right\|}{\|\mathbf{x}^{\delta}\|},$$

for the nonstationary algorithms for both the IT and GIT.

Baart We consider the example `baart` and fix $\xi = 0.01$. Figure 4.2(a) shows the desired solution $\mathbf{x}^{\dagger}$, a uniform sampling of $\sin(t)$ with $t \in [0, \pi]$, and the right-hand side $\mathbf{b}^{\delta}$. Consider first stationary iterated Tikhonov. Figure 4.1(a) shows the RRE for computed solutions determined by the discrepancy principle for $L = I$, $L = L_1$, and $L = L_2$. The regularization parameter $\alpha > 0$ has to be chosen differently for the different regularization matrices. For instance, α has to be chosen much larger for $L = L_2$ than for $L = I$. This is due to the fact that $\mathbf{x}^{\dagger}$ has a large component in $\mathcal{N}(L_2)$. Therefore, α has to be fairly large to make the penalty term $\alpha \|L_2\mathbf{x}\|$ effective. We remark that Algorithm 4.1 converges for any $\alpha > 0$, but the rate



(a) (b)

FIGURE 4.2: `Baart` test problem: (a) desired solution $\mathbf{x}^\dagger$ (dashed curve) and error-contaminated data vector $\mathbf{b}^\delta$ (solid curve), (b) Reconstructions obtained with the nonstationary iterated Tikhonov method with $L = I$ (dashed curve), with $L = L_1$ (solid gray curve), and with $L = L_2$ (solid black curve). The dotted curve shows the desired solution $\mathbf{x}^\dagger$.

of convergence is affected by the choice of α. Choosing α in a proper range, we observe a substantial reduction of the RRE when using GIT with L_1 and, in particular with L_2, when compared with $L = I$. We set the maximum number of iterations to 10^4. Large values of α did not result in accurate approximations of $\mathbf{x}^\dagger$ within this number of iterations.

For the sake of completeness, we show the number of iterations needed for each tested value of α in Figure 4.3(a). We see that the number of iterations needed to satisfy the discrepancy principle increases with α. For α sufficiently large, Algorithm 4.1 terminates because the maximum number of iterations, 10^4, has been reached. For the regularization matrix L_2, a large value of α is required for the regularization term $\alpha \|L\mathbf{x}\|^2$ to be effective (see Figure 4.1(a)). Therefore, the tested α values are not large enough to show a significant increase in the number of iterations.

We would like to mention that the qualitative behavior of the curves in Figure 4.1 does not depend on the noise level. For instance, consider the `baart` example with noise level $\xi = 0.05$ and apply the GIT algorithm with $L \in \{I, L_1, L_2\}$ for α values in the range $[10^{-7}, 10^7]$. Figure 4.3(b) displays the RRE in the approximated solutions determined by Algorithm 4.1 for the α. Comparing Figure 4.1(a) and 4.3(b) shows the errors in the computed approximate solution to differ for $\xi = 0.01$ and $\xi = 0.05$; the computed approximate solutions determined for $\xi = 0.01$ are more accurate. However, the qualitative behavior of the curves is similar.

In the following examples we will not show plots analogous to those of Figure 4.3 because are quite similar.

We turn to the nonstationary iterations. Comparing the RREs in Table 4.1, we can see that both $L = L_1$ and $L = L_2$ yield more accurate approximations of $\mathbf{x}^\dagger$ than $L = I$. This is also confirmed by visual inspection of the computed solutions in Figure 4.2(b). Table 4.1 shows that the components of the computed solutions in $\mathcal{N}(A)$ are small for the GIT$_{NS}$ methods. Their size depends on the matrix L. This is to be expected since the presence of a component $\mathcal{N}(A)$ is due to L. We obtain a much smaller component in $\mathcal{N}(A)$ for L_1 than for L_2. Nevertheless, the latter regularization matrix gives a more accurate approximation of $\mathbf{x}^\dagger$.

We remark that the dimension of the numerical null space of A is very large, about 990. This may contribute to the fact that the computed solutions do not have negligible components

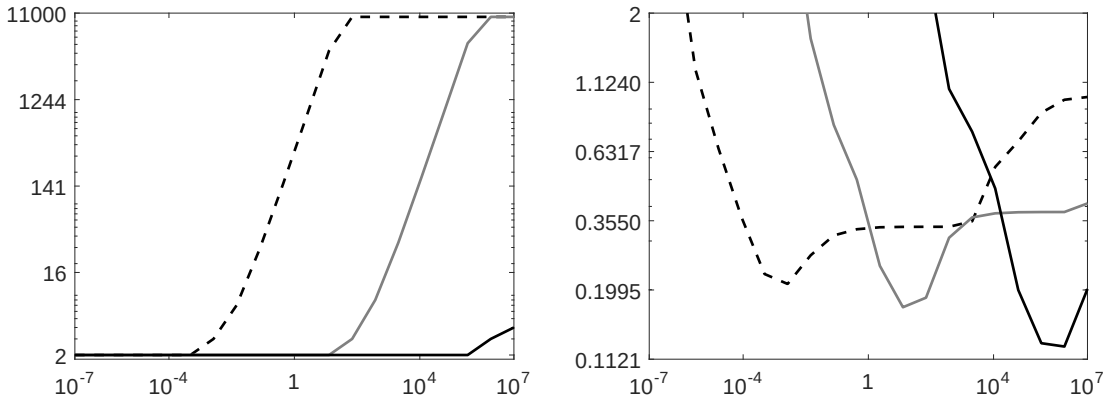


FIGURE 4.3: Baart test problem: (a) Number of iterations prescribed by the discrepancy principle using GIT with $\xi = 0.01$ as a function of α, (b) RRE for the iterates determined by the discrepancy principle using GIT with $\xi = 0.05$ for different values of α. The dashed curves are for $L = I$, the solid gray curves for $L = L_1$, and the solid black curves for $L = L_2$.

Method	α_0	RRE	Iterations	$\frac{\|P_{\mathcal{N}(A)}\mathbf{x}^\delta\|}{\|\mathbf{x}^\delta\|}$
IT_{NS}	10^{-2}	0.17131	4	1.7815×10^{-15}
GIT_{NS} L_1	10^{2}	0.12331	3	9.1999×10^{-15}
GIT_{NS} L_2	10^{6}	**0.04290**	2	0.0027300

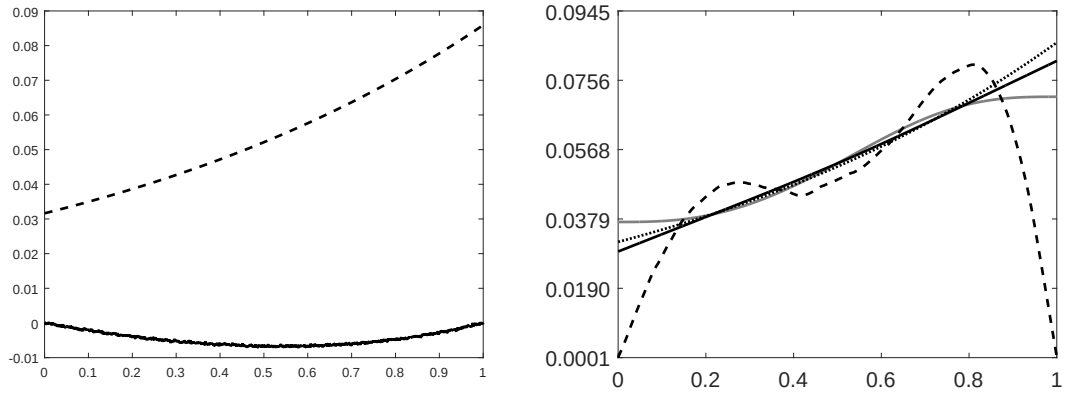
TABLE 4.1: `Baart` test problem: RRE, number of iterations, and relative magnitude of $P_{\mathcal{N}(A)}\mathbf{x}^\delta$ for the nonstationary iterated Tikhonov method with $L = I$ (IT_{NS}), and with $L = L_1$ and L_2 (GIT_{NS}). The sequence of α_k is defined by (2.20) with α_0 shown in the table and $q = 0.8$ for all methods. The smallest error is shown in boldface.

in $\mathcal{N}(A)$. The matrices A in the following examples in one-space dimension have numerical null spaces of much smaller dimension, and the computed approximate solutions have a much smaller component in $\mathcal{N}(A)$. We finally note that the IT_{NS} method yields a negligible component in $\mathcal{N}(A)$.

Deriv2 We now consider the example `deriv2` with $\xi = 0.05$. Figure 4.4(a) displays the desired solution $\mathbf{x}^\dagger$ and the data vector $\mathbf{b}^\delta$. The vector $\mathbf{x}^\dagger$ is a uniform sampling of the function e^t with $t \in [0, 1]$.

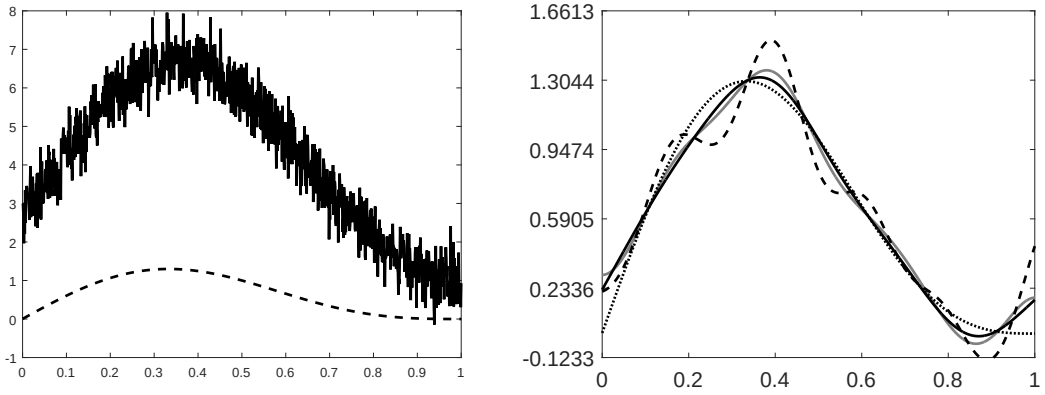
Figure 4.1(b) shows results for the stationary iterated Tikhonov method. The results are comparable to those of the previous example, but the range of α-values that yield reasonably fast convergence is smaller. A proper estimation of α can be avoided by using nonstationary iterated Tikhonov methods. For the latter methods $L = L_1$ and $L = L_2$ yield approximate solutions of higher quality than $L = I$; see Table 4.2 as well as Figure 4.4(b). The regularization matrix L_2 gives the best result. Table 4.2 shows that for all methods the computed approximate solutions have a negligible component in $\mathcal{N}(A)$.

Gravity The last example in one space-dimension is `gravity`. We add white Gaussian noise to the error-free data vector $\mathbf{b}$ to determine an error-contaminated data vector $\mathbf{b}^\delta$ with



(a) (b)

FIGURE 4.4: Deriv2 test problem: (a) desired solution $\mathbf{x}^\dagger$ (dashed curve) and error-contaminated data vector $\mathbf{b}^\delta$ (solid curve), (b) Reconstructions obtained with the nonstationary iterated Tikhonov method with $L = I$ (dashed curve), with $L = L_1$ (solid gray curve), and with $L = L_2$ (solid black curve). The dotted curve shows the desired solution $\mathbf{x}^\dagger$.

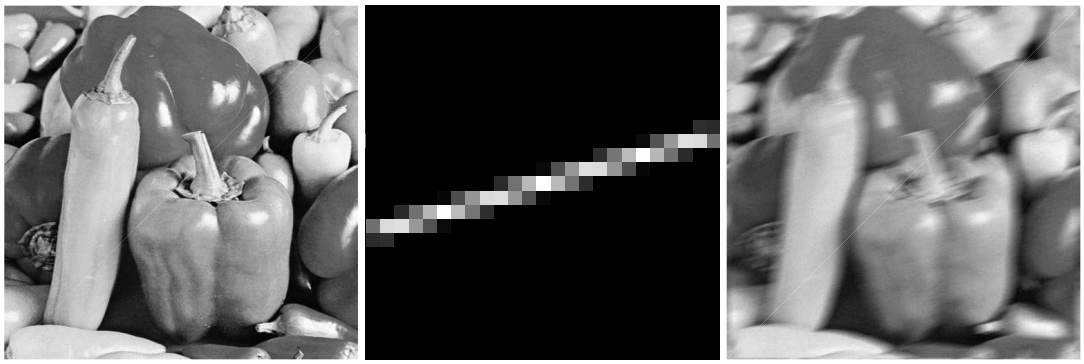


(a) (b)

FIGURE 4.5: `Gravity` test problem: (a) desired solution $\mathbf{x}^\dagger$ (dashed curve) and error-contaminated data vector $\mathbf{b}^\delta$ (solid curve), (b) Reconstructions obtained with the nonstationary iterated Tikhonov method with $L = I$ (dashed curve), with $L = L_1$ (solid gray curve), and with $L = L_2$ (solid black curve). The dotted curve shows the desired solution $\mathbf{x}^\dagger$.

Method	α_0	RRE	Iterations	$\frac{\|P_{\mathcal{N}(A)}\mathbf{x}^\delta\|}{\|\mathbf{x}^\delta\|}$
IT_{NS}	10^{-2}	0.32502	18	2.9408×10^{-15}
GIT_{NS} L_1	10^{2}	0.07138	5	2.8801×10^{-15}
GIT_{NS} L_2	10^{6}	**0.02748**	2	2.8411×10^{-15}

TABLE 4.2: `Deriv2` test problem: RRE, number of iterations, and relative magnitude of $P_{\mathcal{N}(A)}\mathbf{x}^\delta$ for the nonstationary iterated Tikhonov method with $L = I$ (IT_{NS}), and with $L = L_1$ and L_2 (GIT_{NS}). The sequence of α_k is defined by (2.20) with α_0 shown in the table and $q = 0.8$ for all methods. The smallest error is shown in boldface.



(a) (b) (c)

FIGURE 4.6: Peppers test problem: (a) Uncontaminated image (512×512 pixels), (b) PSF (25×25 pixels), (c) blur- and noise-contaminated image ($\xi = 0.03$).

Method	α_0	RRE	Iterations	$\frac{\|P_{\mathcal{N}(A)}\mathbf{x}^\delta\|}{\|\mathbf{x}^\delta\|}$
IT_{NS}	10^{-2}	0.17001	2	4.1708×10^{-15}
GIT_{NS} L_1	10^2	0.10165	2	1.4004×10^{-9}
GIT_{NS} L_2	10^6	**0.081483**	2	6.4620×10^{-10}

TABLE 4.3: `Gravity` test problem: RRE, number of iterations, and relative magnitude of $P_{\mathcal{N}(A)}\mathbf{x}^\delta$ for the nonstationary iterated Tikhonov method with $L = I$ (IT_{NS}), and with $L = L_1$ and L_2 (GIT_{NS}). The sequence of α_k is defined by (2.20) with α_0 shown in the table and $q = 0.8$ for all methods. The smallest error is shown in boldface.

$\xi = 0.1$. The desired solution, $\mathbf{x}^\dagger$, is a uniform sampling of $\sin(\pi t) + \frac{1}{2}\sin(2\pi t)$ with $t \in [0, 1]$. Both $\mathbf{x}^\dagger$ and $\mathbf{b}^\delta$ are displayed in Figure 4.5(a).

Figure 4.1(c) shows the RRE values at termination for different α-values for stationary iterated Tikhonov methods. The graphs are similar as for the previous examples. Table 4.3 compares RREs obtained for nonstationary iterated Tikhonov methods. We observe that all nonstationary methods in our comparison converge in only 2 iterations. This is due to the large amount of noise in $\mathbf{b}^\delta$. The more error in $\mathbf{b}^\delta$, the faster the discrepancy principle is satisfied. Similarly as in the previous examples, we see that the use of a regularization matrix different from the identity is beneficial; see Figure 4.5(b). In particular, the approximations of $\mathbf{x}^\dagger$ obtained with GIT_{NS} are smooth despite the high noise level. Looking at the component of the solution in $\mathcal{N}(A)$, we can see that is very small.

Peppers Our last example illustrates the application of Algorithm 4.2 to an image deblurring problem. The peppers image in Figure 4.6(a) represents the blur- and noise-free image. The blurred image is constructed by blurring the exact image by motion blur defined by the point-spread function (PSF) shown in Figure 4.6(b). We add white Gaussian noise such that $\xi = 0.03$ to the blurred image. This gives the blur- and noise-contaminated image in Figure 4.6(c). We ignore boundary effects and use convolution with periodic boundary conditions to define A.

We use regularization matrices that are a scaled discretizations of periodic divergence L_1 defined in (2.14) or a scaled discretization of the periodic Laplacian L_2. We define

$$L_2 = L_2^1 \otimes I + I \otimes L_2^1, \tag{4.17}$$

(a) (b) (c)

FIGURE 4.7: Peppers test problem reconstructions determined by the nonstationary iterated Tikhonov method with (a) $L = I$, (b) $L = L_1$, and (c) $L = L_2$.

Method	RRE	Iterations
IT_{NS}	0.10743	7
GIT_{NS} L_1	0.09368	4
GIT_{NS} L_2	**0.08516**	3

TABLE 4.4: Peppers test problem: RRE and number of iterations for the nonstationary iterated Tikhonov method with $L = I$ (IT_{NS}), and with $L = L_1$ and L_2 (GIT_{NS}). The sequence of α_k is defined by (2.20) with $\alpha_0 = 1$ and $q = 0.8$ for all methods. The smallest error is shown in boldface.

where

$$L_2^1 = \begin{pmatrix} 2 & -1 & & & -1 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ -1 & & & -1 & 2 \end{pmatrix}$$

denotes the discretization of the second derivative in one space-dimension with periodic boundary conditions. Both L_1 and L_2 are BCCB (block circulant with circulant block) matrices and therefore can be diagonalized using the 2D discrete Fourier transform.

We first consider the stationary iterated Tikhonov method. Figure 4.1(d) displays the RRE of the approximate solution determined by using the discrepancy principle for different values of α. We get stagnation for large α-values. Moreover, for every $\alpha > 0$, the stationary iterated Tikhonov method with L given by (2.14) or (4.17) gives better results than with $L = I$ for the same α-value.

Turning to the nonstationary iterated Tikhonov method, Table 4.4 illustrates that the use of the regularization matrices L_1 and L_2 give smaller errors in the computed approximate solutions than when the identity matrix is used as regularization matrix. Figure 4.7 shows that the regularization matrices L_1 and L_2 give restorations with less "ringing" and with sharper edges than when using the identity as regularization matrix.

Chapter 5

Fractional and Weighted Iterated Tikhonov

In this and in the next chapter we consider the more general framework of linear operator equations of the form

$$Ax = b\,, \tag{5.1}$$

where $A : \mathcal{X} \to \mathcal{Y}$ is a compact linear operator between Hilbert spaces $\mathcal{X}$ and $\mathcal{Y}$. We assume b to be attainable, i.e., that problem (5.1) has a solution $x^\dagger = A^\dagger b$ of minimal norm. $A^\dagger$ is unbounded because A is compact, with infinite dimensional range. Hence problem (5.1) is ill-posed and has to be regularized.

Like before only an approximation b^δ of b is available with

$$\|b^\delta - b\| \leq \delta. \tag{5.2}$$

As in the finite dimensional case $A^\dagger b^\delta$ is not a good approximation of $x^\dagger$, thus we approximate $x^\dagger$ with $x_\alpha^\delta := R_\alpha b^\delta$ where $\{R_\alpha\}$ is a family of continuous operators depending on a parameter α that will be defined later. In this setting the standard Tikhonov regularization is defined by $R_\alpha = (A^*A + \alpha I)^{-1} A^*$.

Using the singular values expansion of A, filter based regularization methods are defined in terms of filters of the singular values, cf. Proposition 5.3. This is a useful tool for the analysis of regularization techniques [79], both for direct and iterative regularization methods [80, 84]. Furthermore, new regularization methods can be defined investigating new classes of filters. We are going to consider two different variants of standard Tikhonov regularization stemming from this interpretation. The first has been proposed in [95] and it is called fractional Tikhonov method. The authors obtain a new class of filtering regularization methods adding an exponent, depending on a parameter, to the filter of the standard Tikhonov method. They provide a detailed analysis of the filtering properties and the optimal order of the method in terms of such further parameter. The second and different generalization of the standard Tikhonov method we are going to consider has been recently proposed in [86] with a detailed filtering analysis. Both generalizations are called "fractional Tikhonov regularization" in the literature and they are compared in [69], where the optimal order of the method in [86] is provided as well. To distinguish the two proposals in [95] and [86], we will refer in the following as "fractional Tikhonov regularization" and "weighted Tikhonov regularization", respectively. These variants of the Tikhonov method have been introduced to compute accurate approximations of non-smooth solutions, since it is well known that the Tikhonov method provides over-smoothed solutions.

In this chapter, we firstly provide a saturation result similar to the well-known saturation result for Tikhonov regularization [61]: indeed, Tikhonov regularization under suitable a-priori assumption and a-priori choice rule, $\alpha = \alpha(\delta) \sim c(\delta)^{2/3}$, is of optimal order and the best possible convergence rate obtainable is

$$\|x_\alpha^\delta - x^\dagger\| = O\left(\delta^{\frac{2}{3}}\right).$$

On the other hand, let $\mathcal{R}(A)$ be the range of A and let Q be the orthogonal projector onto $\overline{\mathcal{R}(A)}$, if

$$\sup\left\{\|x_\alpha^\delta - x^\dagger\| : \|Q\left(b - b^\delta\right)\| \leq \delta\right\} = o\left(\delta^{\frac{2}{3}}\right),$$

then $x^\dagger = 0$, as long as $\mathcal{R}(A)$ is not closed, and this shows how Tikhonov regularization for an ill-posed problem with compact operator never yields a convergence rate which is faster than $O\left(\delta^{\frac{2}{3}}\right)$, since it saturates at this rate. Such result motivated us to introduce the iterated version of fractional and weighted Tikhonov in the same spirit of the iterated Tikhonov method, see Section 2.2.2. We prove that those iterated methods can overcome the previous saturation results. Afterwards, inspired by the works [25, 79] we introduce the nonstationary variants of our iterated methods. Differently from the nonstationary iterated Tikhonov, we have two nonstationary sequences of parameters. In the noise free case, we give sufficient conditions on these sequences to guarantee the convergence providing also the corresponding convergence rates. In the noise case, we show the stability of the proposed iterative schemes proving that they are regularization methods. Finally, a few selected examples confirm the previous theoretical analysis, showing that a proper choice of the nonstationary sequences of parameters can provide better restorations compared to the classical iterated Tikhonov with a geometric sequence of regularization parameter according to [25].

This chapter is structured as follows: in Section 5.1 we first recall the basic definition of filter based regularization methods and of optimal order of a regularization method. Then in Section 5.2 Fractional Tikhonov regularization with optimal order and converse results are studied and we provide saturation results for both. We then introduce, in Section 5.3 new iterated fractional Tikhonov regularization methods are introduced, where the analysis of their convergence rate shows that they are able to overcome the previous saturation results. A nonstationary iterated weighted Tikhonov regularization and a similar nonstationary iterated fractional Tikhonov regularization are then investigated in detail in Section 5.4. Finally, we give some numerical examples in Section 5.5.

5.1 Preliminaries

As described above, we consider a compact linear operator $A : \mathcal{X} \to \mathcal{Y}$ between Hilbert spaces $\mathcal{X}$ and $\mathcal{Y}$ (over the field $\mathbb{R}$ or $\mathbb{C}$) with given inner products $\langle\cdot,\cdot\rangle_\mathcal{X}$ and $\langle\cdot,\cdot\rangle_\mathcal{Y}$, respectively. Hereafter we will omit the subscript for the inner product as it will be clear in the context. If $A^* : \mathcal{Y} \to \mathcal{X}$ denotes the adjoint of A (i.e., $\langle Ax, y\rangle = \langle x, A^*y\rangle$), then we indicate with $(\sigma_j; v_j, u_j)_{n\in\mathbb{N}}$ the singular value expansion of A, where $\{v_j\}_{j\in\mathbb{N}}$ and $\{u_j\}_{j\in\mathbb{N}}$ are a complete orthonormal system of eigenvectors for A^*A and AA^*, respectively, and $\sigma_j > 0$ are written in decreasing order, with 0 being the only accumulating point for the sequence $\{\sigma_j\}_{j\in\mathbb{N}}$ when $\dim \mathcal{R}(A) = \infty$. If $\mathcal{X}$ is not finite dimensional, then $0 \in \lambda(A^*A)$, the spectrum of A^*A, namely $\lambda(A^*A) = \{0\} \cup \bigcup_{j=1}^\infty \{\sigma_j^2\}$. Finally, $\sigma(A)$ denotes the closure of $\bigcup_{j=1}^\infty \{\sigma_j\}$, i.e., $\sigma(A) = \{0\} \cup \bigcup_{j=1}^\infty \{\sigma_j\}$.

Let now $\{E_{\sigma^2}\}_{\sigma^2\in\lambda(A^*A)}$ be the spectral decomposition of the self-adjoint operator A^*A. Then from well-known facts from functional analysis [116] we can write $f\left(A^*A\right) := \int f\left(\sigma^2\right) \mathrm{d}E_{\sigma^2}$, where $f : \lambda\left(A^*A\right) \subset \mathbb{R} \to \mathbb{C}$ is a bounded Borel measurable function and $\langle Ex_1, x_2\rangle$ is a regular complex Borel measure for every $x_1, x_2 \in \mathcal{X}$. The following equalities hold

$$Ax = \sum_{l=1}^{+\infty} \sigma_l \langle x, v_l\rangle u_l, \qquad x \in \mathcal{X}, \tag{5.3}$$

$$A^*b = \sum_{l=1}^{+\infty} \sigma_l \langle b, u_l\rangle v_l, \qquad b \in \mathcal{Y}, \tag{5.4}$$

$$f\left(A^*A\right)x := \int_{\lambda(A^*A)} f\left(\sigma^2\right) \mathrm{d}E_{\sigma^2}x = \sum_{l=1}^{\infty} f\left(\sigma_l^2\right) \langle x, v_l\rangle v_l,$$

$$\langle f\left(A^*A\right)x_1, x_2\rangle = \int_{\lambda(A^*A)} f\left(\sigma^2\right) \mathrm{d}\langle E_{\sigma^2}x_1, x_2\rangle = \sum_{l=1}^{\infty} f\left(\sigma_l^2\right) \overline{\langle b, v_l\rangle}\langle x, v_l\rangle,$$

$$\|f\left(A^*A\right)\| = \sup\{|f\left(\sigma^2\right)| : \sigma^2 \in \lambda\left(A^*A\right)\},$$

where the series (5.3) and (5.4) converge in the L^2 norms induced by the scalar products of $\mathcal{X}$ and $\mathcal{Y}$, respectively.

Definition 5.1. *We define the* generalized inverse $A^\dagger : \mathcal{D}\left(A^\dagger\right) \subseteq \mathcal{Y} \to \mathcal{X}$ *of a compact linear operator* $A : \mathcal{X} \to \mathcal{Y}$ *as*

$$A^\dagger b = \sum_{l:\,\sigma_l>0} \sigma_l^{-1} \langle b, u_l\rangle v_l, \qquad b \in \mathcal{D}\left(A^\dagger\right), \tag{5.5}$$

where

$$\mathcal{D}\left(A^\dagger\right) = \left\{ b \in \mathcal{Y} : \sum_{l:\,\sigma_l>0} \sigma_l^{-2} |\langle b, u_l\rangle|^2 < \infty \right\}.$$

With respect to problem (5.1), we consider the case where only an approximation b^δ of b satisfying the condition (5.2) is available. Therefore $x^\dagger = A^\dagger b$, $b \in \mathcal{D}\left(A^\dagger\right)$, cannot be approximated by $A^\dagger b^\delta$, due to the unboundedness of $A^\dagger$, and hence in practice the problem (5.1) is approximated by a family of neighboring well-posed problems [61].

Definition 5.2. *By a* regularization method *for* $A^\dagger$ *we call any family of operators*

$$\{R_\alpha\}_{\alpha\in(0,\alpha_0)} : \mathcal{Y} \to \mathcal{X}, \qquad \alpha_0 \in (0, +\infty],$$

with the following properties:

(i) $R_\alpha : \mathcal{Y} \to \mathcal{X}$ *is a bounded operator for every* α.

(ii) For every $b \in \mathcal{D}\left(A^\dagger\right)$ *there exists a mapping (rule choice)* $\alpha : \mathbb{R}_+ \times \mathcal{Y} \to (0, \alpha_0) \in \mathbb{R}$, $\alpha = \alpha\left(\delta, b^\delta\right)$, *such that*

$$\limsup_{\delta\to 0} \left\{ \alpha\left(\delta, b^\delta\right) : b^\delta \in \mathcal{Y}, \|b - b^\delta\| \le \delta \right\} = 0,$$

and

$$\limsup_{\delta\to 0} \left\{ \|R_{\alpha\left(\delta,b^\delta\right)} b^\delta - A^\dagger b\| : b^\delta \in \mathcal{Y}, \|b - b^\delta\| \le \delta \right\} = 0.$$

Throughout this chapter c is a constant which can change from one instance to the next.

For the sake of clarity, if more than one constant will appear in the same equation we will distinguish them by means of a subscript.

Proposition 5.3. *Let $A : \mathcal{X} \to \mathcal{Y}$ be a compact linear operator and $A^\dagger$ its generalized inverse. Let $R_\alpha : \mathcal{Y} \to \mathcal{X}$ be a family of operators defined for every $\alpha \in (0, \alpha_0)$ as*

$$R_\alpha b := \sum_{l:\, \sigma_l > 0} F_\alpha(\sigma_l)\, \sigma_l^{-1} \langle b, u_l \rangle v_l, \tag{5.6}$$

where $F_\alpha : [0, \sigma_1] \supset \sigma(A) \to \mathbb{R}$ is a Borel function such that

$$\sup_{l:\, \sigma_l > 0} |F_\alpha(\sigma_l)\, \sigma_l^{-1}| = c(\alpha) < \infty, \tag{5.7}$$

$$|F_\alpha(\sigma_l)| \le c < \infty, \qquad \text{where } c \text{ does not depend on } (\alpha, l), \tag{5.8}$$

$$\lim_{\alpha \to 0} F_\alpha(\sigma_l) = 1 \ \text{ point-wise in } \sigma_l. \tag{5.9}$$

Then R_α is a regularization method, with $\|R_\alpha\| = c(\alpha)$, and it is called filter based regularization method.

Proof. See [96] and [61]. □

For the sake of notational brevity, we fix the following notation

$$x_\alpha := R_\alpha b, \qquad b \in \mathcal{D}\left(A^\dagger\right), \tag{5.10}$$

$$x_\alpha^\delta := R_\alpha b^\delta, \qquad b^\delta \in \mathcal{Y}. \tag{5.11}$$

We report hereafter the definition of optimal order, under the same a-priori assumption given in [61].

Definition 5.4. *For every given $\nu, \rho > 0$, let*

$$\mathcal{X}_{\nu,\rho} := \left\{ x \in \mathcal{X} : \exists\, \omega \in \mathcal{X}, \|\omega\| \le \rho, x = (A^* A)^{\frac{\nu}{2}} \omega \right\} \subset \mathcal{X}.$$

A regularization method R_α is said to be of optimal order *under the a-priori assumption $x^\dagger \in \mathcal{X}_{\nu,\rho}$ if*

$$\Delta(\delta, \mathcal{X}_{\nu,\rho}, R_\alpha) \le c \delta^{\frac{\nu}{\nu+1}} \rho^{\frac{1}{\nu+1}}, \tag{5.12}$$

where for any general set $M \subseteq X$, $\delta > 0$ and for a regularization method R_α, we define

$$\Delta(\delta, M, R_\alpha) := \sup \left\{ \|x^\dagger - x_\alpha^\delta\| : x^\dagger \in M,\ \|b - b^\delta\| \le \delta \right\}.$$

If ρ is not known, as it will be usually the case, then we relax the definition introducing the set

$$\mathcal{X}_\nu := \bigcup_{\rho > 0} \mathcal{X}_{\nu,\rho}$$

and saying that a regularization method R_α is called of optimal order *under the a-priori assumption $x^\dagger \in \mathcal{X}_\nu$ if*

$$\Delta(\delta, \mathcal{X}_\nu, R_\alpha) \le c \delta^{\frac{\nu}{\nu+1}}. \tag{5.13}$$

Remark 5.5. *Since we are concerned with the rate with which $\|x^\dagger - x_\alpha^\delta\|$ converges to zero as $\delta \to 0$, the a-priori assumption $x^\dagger \in \mathcal{X}_\nu$ is usually sufficient for the optimal order analysis, requiring that* (5.13) *is satisfied.*

Hereafter we cite a theorem which states sufficient conditions for order optimality, when filtering methods are employed, see [96, Proposition 3.4.3, pag. 58].

Theorem 5.6. *[96] Let $A : \mathcal{X} \to \mathcal{Y}$ be a compact linear operator, ν and $\rho > 0$, and let $R_\alpha : \mathcal{Y} \to \mathcal{X}$ be a filter based regularization method. If there exists a fixed $\beta > 0$ such that*

$$\sup_{0<\sigma\leq\sigma_1} |F_\alpha(\sigma)\,\sigma^{-1}| \leq c\alpha^{-\beta}, \tag{5.14a}$$

$$\sup_{0\leq\sigma\leq\sigma_1} |(1 - F_\alpha(\sigma))\,\sigma^{\nu}| \leq c_\nu\alpha^{\beta\nu}, \tag{5.14b}$$

then R_α is of optimal order, under the a-priori assumption $x^\dagger \in \mathcal{X}_{\nu,\rho}$, with the choice rule

$$\alpha = \alpha(\delta,\rho) = \eta\left(\frac{\delta}{\rho}\right)^{\frac{1}{\beta(\nu+1)}}, \qquad 0 < \eta = \left(\frac{c}{\nu c_\nu}\right)^{\frac{1}{\beta(\nu+1)}}.$$

If we are concerned just about the rate of convergence with respect to only δ, the preceding theorem can be applied under the a-priori assumption $x^\dagger \in X_\nu$, fitting the proof to the latter case without any effort. On the contrary, below we present a converse result.

Theorem 5.7. *Let A be a compact linear operator with infinite dimensional range and let R_α be a filter based regularization method with filter function $F_\alpha : [0,\sigma_1] \supset \sigma(A) \to \mathbb{R}$. If there exist ν and $\beta > 0$ such that*

$$(1 - F_\alpha(\sigma))\,\sigma^{\nu} \geq c\alpha^{\beta\nu} \qquad \text{for } \sigma \in [c'\alpha^\beta, \sigma_1] \tag{5.15}$$

and

$$\|x^\dagger - x_\alpha\| = O\left(\alpha^{\beta\nu}\right), \tag{5.16}$$

then $x^\dagger \in \mathcal{X}_\nu$.

Proof. By (5.5) and (5.6), it holds

$$\begin{aligned}
\|x^\dagger - x_\alpha\|^2 &= \sum_{\sigma_l>0} (1 - F_\alpha(\sigma_l))^2\,\sigma_l^{-2}|\langle b, u_l\rangle|^2 \\
&= \sum_{\sigma_l>0} (1 - F_\alpha(\sigma_l))^2\,|\langle x^\dagger, v_l\rangle|^2 \\
&= \sum_{\sigma_l>0} [(1 - F_\alpha(\sigma_l))\,\sigma_l^{\nu}]^2\,\sigma_l^{-2\nu}|\langle x^\dagger, v_l\rangle|^2 \\
&\geq \left(c\alpha^{\beta\nu}\right)^2 \sum_{\sigma_l\geq c'\alpha^\beta} \sigma_l^{-2\nu}|\langle x^\dagger, v_l\rangle|^2,
\end{aligned}$$

thanks to the assumption (5.15). From (5.16) we deduce that

$$\lim_{\alpha^\beta\to 0} \sum_{\sigma_l\geq c'\alpha^\beta} \sigma_l^{-2\nu}|\langle x^\dagger, v_l\rangle|^2 < +\infty.$$

Finally, if we define $\omega := \sum_{\sigma_l>0} \sigma^{-\nu}\langle x^\dagger, v_l\rangle v_l$, then ω is well defined and $(A^*A)^{\nu/2}\,\omega = x^\dagger$, i.e., $x^\dagger \in X_\nu$. ∎

5.2 Fractional variants of Tikhonov regularization

In this section we discuss two recent types of regularization methods that generalize the classical Tikhonov method and that were first introduced and studied in [86] and [95].

5.2.1 Weighted Tikhonov regularization

Definition 5.8 ([86]). *We call* Weighted Tikhonov *method the filter based method*

$$R_{\alpha,r}b := \sum_{l:\,\sigma_l>0} F_{\alpha,r}\left(\sigma_l\right)\sigma_l^{-1}\langle b, u_l\rangle v_l,$$

where the filter function is

$$F_{\alpha,r}\left(\sigma\right) = \frac{\sigma^{r+1}}{\sigma^{r+1}+\alpha}, \tag{5.17}$$

for $\alpha > 0$ and $r \geq 0$.

According to (5.10) and (5.11), we fix the following notation

$$x_{\alpha,r} := R_{\alpha,r}b, \qquad b \in \mathcal{D}\left(A^\dagger\right), \tag{5.18}$$

$$x^\delta_{\alpha,r} := R_{\alpha,r}b^\delta, \qquad b^\delta \in \mathcal{Y}. \tag{5.19}$$

Remark 5.9. *The Weighted Tikhonov method can also be defined as the unique minimizer of the following functional,*

$$R_{\alpha,r}b := \operatorname{argmin}_{x\in X}\left\{\|Ax-b\|^2_W + \alpha\|x\|^2\right\}, \tag{5.20}$$

where the semi-norm $\|\cdot\|_W$ is induced by the operator $W := (AA^*)^{\frac{r-1}{2}}$. For $0 \leq r < 1$, W is to be intended as the Moore-Penrose (pseudo) inverse. Developing the calculations, it follows that

$$R_{\alpha,r}b = \left[(A^*A)^{\frac{r+1}{2}} + \alpha I\right]^{-1}(A^*A)^{\frac{r-1}{2}}A^*b. \tag{5.21}$$

That is the reason that motivated us to rename the original method that appeared in [86], as weighted Tikhonov method. *In this way it would be easier to distinguish it from the* fractional Tikhonov method *introduced in [95].*

The optimal order of the weighted Tikhonov regularization was proved in [69]. The following proposition restates such result, putting in evidence the dependence on r of ν, and provides a converse result.

Proposition 5.10. *Let A be a compact linear operator with infinite dimensional range. For every given $r \geq 0$ the weighted Tikhonov method, $R_{\alpha,r}$, is a regularization method of optimal order, under the a-priori assumption $x^\dagger \in X_{\nu,\rho}$, with $0 < \nu \leq r+1$. The best possible rate of convergence with respect to δ is $\|x^\dagger - x^\delta_{\alpha,r}\| = O\left(\delta^{\frac{r+1}{r+2}}\right)$, that is obtained for $\alpha = \left(\frac{\delta}{\rho}\right)^{\frac{r+1}{\nu+1}}$ with $\nu = r+1$. On the other hand, if $\|x^\dagger - x_{\alpha,r}\| = O\left(\alpha\right)$ then $x^\dagger \in \mathcal{X}_{r+1}$.*

Proof. For weighted Tikhonov the left-hand side of condition (5.14a) becomes

$$\sup_{0<\sigma\leq\sigma_1}\left|\frac{\sigma^r}{\sigma^{r+1}+\alpha}\right|.$$

By derivation, if $r > 0$ then it is straightforward to see that the quantity above is bounded by $\alpha^{-\beta}$, with $\beta = 1/(r+1)$. Similarly, the left-hand side of condition (5.14b) takes the form

$$\sup_{0\le\sigma\le\sigma_1} \left| \frac{\alpha\sigma^\nu}{\sigma^{r+1}+\alpha} \right|,$$

and it is easy to check that it is bounded by $\alpha^{\beta\nu}$ if and only if $0 < \nu \le r+1$. From Theorem 5.6, as long as $0 < \nu \le r+1$, with $r > 0$, if $x^\dagger \in \mathcal{X}_{\nu,\rho}$ then we find order optimality (5.12) and the best possible rate of convergence obtainable with respect to δ is $O\left(\delta^{\frac{r+1}{\nu+1}}\right)$, for $\nu = r+1$.

On the contrary, with $\beta = 1/(r+1)$ and $\nu = r+1$, we deduce that

$$|(1 - F_{\alpha,r}(\sigma))\,\sigma^\nu| = \frac{\alpha\sigma^\nu}{\sigma^{r+1}+\alpha} \ge \frac{1}{2}\alpha, \qquad \text{for } \sigma \in [\alpha^\beta, \sigma_1].$$

Therefore, if $\|x^\dagger - x_{\alpha,r}\| = O(\alpha)$ then $x^\dagger \in \mathcal{X}_\nu$ by Theorem 5.7. □

The following proposition deals with a saturation result similar to a well known result for classic Tikhonov, cf. [61, Proposition 5.3].

Proposition 5.11 (Saturation for weighted Tikhonov regularization)**.** *Let $A : \mathcal{X} \to \mathcal{Y}$ be a compact linear operator with infinite dimensional range and $R_{\alpha,r}$ be the corresponding family of weighted Tikhonov regularization operators in Definition 5.8. Let $\alpha = \alpha\left(\delta, b^\delta\right)$ be any parameter choice rule. If*

$$\sup\left\{ \|x^\delta_{\alpha,r} - x^\dagger\| : \|Q\left(b - b^\delta\right)\| \le \delta \right\} = o\left(\delta^{\frac{r+1}{r+2}}\right), \tag{5.22}$$

then $x^\dagger = 0$, where Q denotes the orthogonal projector onto $\overline{R(A)}$.

Proof. Define

$$\delta_l := \sigma_l^{r+2}, \qquad b_l^\delta := b + \delta_l u_l \text{ so that } \|b - b_l^\delta\| \le \delta_l,$$
$$\alpha_l := \alpha\left(\delta_l, b_l^\delta\right), \qquad x_l := x_{\alpha_l,r}, \qquad x_l^\delta := x^{\delta_l}_{\alpha_l,r}.$$

By the assumption that A has not finite dimensional range, we deduce that $\lim_{l\to\infty}\sigma_l = 0$. According to Remark 5.9, from equation (5.21) we have

$$x_l^\delta - x^\dagger = R_{\alpha_l,r}b_l^\delta - x^\dagger = R_{\alpha_l,r}b + \delta_l R_{\alpha_l,r}u_l - x^\dagger = x_l - x^\dagger + \delta_l F_{\alpha_l,r}(\sigma_l)\,\sigma_l^{-1}v_l$$

and hence by (5.17)

$$\|x_l^\delta - x^\dagger\|^2 = \|x_l - x^\dagger\|^2 + 2\frac{\delta_l\sigma_l^r}{\sigma_l^{r+1}+\alpha_l}\operatorname{Re}\langle x_l - x^\dagger, v_l\rangle + \left(\frac{\delta_l\sigma_l^r}{\sigma_l^{r+1}+\alpha_l}\right)^2.$$

From the choice of $\delta_l := \sigma_l^{r+2}$ follows that

$$\begin{aligned}\left(\delta_l^{-\frac{r+1}{r+2}}\|x_l^\delta - x^\dagger\|\right)^2 &\ge \frac{2}{\delta_l^{\frac{r+1}{r+2}}+\alpha_l}\operatorname{Re}\langle x_l - x^\dagger, v_l\rangle + \left(\frac{\delta_l^{\frac{r+1}{r+2}}}{\delta_l^{\frac{r+1}{r+2}}+\alpha_l}\right)^2 \\ &= \frac{2}{1+\delta_l^{-\frac{r+1}{r+2}}\alpha_l}\delta_l^{-\frac{r+1}{r+2}}\operatorname{Re}\langle x_l - x^\dagger, v_l\rangle + \left(\frac{1}{1+\delta_l^{-\frac{r+1}{r+2}}\alpha_l}\right)^2. \end{aligned} \tag{5.23}$$

By (5.21),

$$\left((A^*A)^{\frac{r+1}{2}} + \alpha_l I\right)\left(x^\dagger - x_l^\delta\right) = (A^*A)^{\frac{r+1}{2}} x^\dagger + \alpha_l x^\dagger - (A^*A)^{\frac{r-1}{2}} A^* b_l^\delta$$
$$= \alpha_l x^\dagger - \delta_l (A^*A)^{\frac{r-1}{2}} A^* u_l,$$

so that

$$\alpha_l \|x^\dagger\| = O\left(\delta_l + \|x^\dagger - x_l^\delta\|\right). \tag{5.24}$$

Since, by assumption, $\|x^\dagger - x_l^\delta\| = o\left(\delta_l^{\frac{r+1}{r+2}}\right)$, it follows from (5.24) that if $x^\dagger \neq 0$, then

$$\lim_{l\to\infty} \alpha_l \delta_l^{-\frac{r+1}{r+2}} = 0. \tag{5.25}$$

Hence, the second term in the right-hand side of (5.23) tends to 1. Since, by assumption, the left-hand side of (5.23) tends to 0, we obtain

$$0 \geq \limsup_{l\to\infty} \frac{2}{1 + \delta_l^{-\frac{r+1}{r+2}} \alpha_l} \delta_l^{-\frac{r+1}{r+2}} \operatorname{Re}\langle x_l - x^\dagger, v_l\rangle + 1.$$

Now, by assumption (5.22), also $\|x_l - x^\dagger\| = o\left(\delta^{\frac{r+1}{r+2}}\right)$, so that, if $x^\dagger \neq 0$, from (5.25) applied to the preceding inequality, we obtain the contradiction $0 \geq 1$. Hence, $x^\dagger = 0$. □

Note that for $r = 1$ (classical Tikhonov) the previous proposition gives exactly Proposition 5.3 in [61]. On the other hand, taking a large r, it is possible to overcome the saturation result of classical Tikhonov obtaining a convergence rate arbitrary close to $O(\delta)$.

5.2.2 Fractional Tikhonov regularization

Here we introduce the *fractional Tikhonov* method defined and discussed in [95].

Definition 5.12 ([95]). *We call* Fractional Tikhonov *method the filter based method*

$$R_{\alpha,\gamma} b := \sum_{l:\,\sigma_l>0} F_{\alpha,\gamma}(\sigma_l)\, \sigma_l^{-1} \langle b, u_l\rangle v_l,$$

where the filter function is

$$F_{\alpha,\gamma}(\sigma) = \frac{\sigma^{2\gamma}}{(\sigma^2 + \alpha)^\gamma},$$

for $\alpha > 0$ and $\gamma \geq 1/2$.

Note that $F_{\alpha,\gamma}$ is well-defined also for $0 < \gamma < 1/2$, but the condition (5.7) requires $\gamma \geq 1/2$ to guarantee that $F_{\alpha,\gamma}$ is a filter function.

We use the notation for $x_{\alpha,\gamma}$ and $x_{\alpha,\gamma}^\delta$ like in equations (5.18) and (5.19), respectively. The optimal order of the fractional Tikhonov regularization was proved in [95, Proposition 3.2]. The following proposition restates such result including also $\gamma = 1/2$ and provides a converse result.

Proposition 5.13. *The extended fractional Tikhonov filter method is a regularization method of optimal order, under the a-priori assumption $x^\dagger \in X_{\nu,\rho}$, for every $\gamma \geq 1/2$ and $0 < \nu \leq 2$. The*

best possible rate of convergence with respect to δ *is* $\|x^\dagger - x^\delta_{\alpha,\gamma}\| = O\left(\delta^{\frac{2}{3}}\right)$, *that is obtained for* $\alpha = \left(\frac{\delta}{\rho}\right)^{\frac{2}{\nu+1}}$ *with* $\nu = 2$. *On the other hand, if* $\|x^\dagger - x_{\alpha,\gamma}\| = O(\alpha)$ *then* $x^\dagger \in \mathcal{X}_2$.

Proof. Condition (5.7) is verified for $\gamma \geq 1/2$ and the same holds for conditions (5.8) and (5.9). Deriving the filter function, it is immediate to see that equation (5.14a) is verified for $\gamma \geq 1/2$, with $\beta = 1/2$. It remains to check equation (5.14b):

$$\begin{aligned}(1 - F_{\alpha,\gamma}(\sigma))\,\sigma^\nu &= \frac{(\sigma^2+\alpha)^\gamma - \sigma^{2\gamma}}{(\sigma^2+\alpha)^\gamma}\sigma^\nu \\ &= \frac{\left(\frac{\sigma^2}{\alpha}+1\right)^\gamma - \left(\frac{\sigma^2}{\alpha}\right)^\gamma}{\left(\frac{\sigma^2}{\alpha}+1\right)^{\gamma-1}} \cdot \frac{\alpha\sigma^\nu}{\sigma^2+\alpha} \\ &= h\left(\frac{\sigma^2}{\alpha}\right) \cdot (1 - F_{\alpha,1}(\sigma))\,\sigma^\nu,\end{aligned}$$

where $h(x) = \frac{(x+1)^\gamma - x^\gamma}{(x+1)^{\gamma-1}}$ is monotone, $h(0) = 1$ for every γ, and $\lim_{x\to\infty} h(x) = \gamma$. Namely $h(x) \in (\gamma, 1]$ for $0 \leq \gamma \leq 1$ and $h(x) \in [1, \gamma)$ for $\gamma \geq 1$. Therefore we deduce that

$$\gamma\,(1 - F_{\alpha,1}(\sigma)) \leq (1 - F_{\alpha,\gamma}(\sigma)) \leq (1 - F_{\alpha,1}(\sigma)), \qquad \text{for } 0 \leq \gamma \leq 1, \tag{5.26}$$

$$(1 - F_{\alpha,1}(\sigma)) \leq (1 - F_{\alpha,\gamma}(\sigma)) \leq \gamma\,(1 - F_{\alpha,1}(\sigma)), \qquad \text{for } \gamma \geq 1, \tag{5.27}$$

from which we infer that

$$\sup_{\sigma\in[0,\sigma_1]} |(1 - F_{\alpha,\gamma}(\sigma))\,\sigma^\nu| \leq \max\{1,\gamma\} \sup_{\sigma\in[0,\sigma_1]} |(1 - F_{\alpha,1}(\sigma))\,\sigma^\nu| \leq c\alpha^{\frac{\nu}{2}},$$

since $F_{\alpha,1}(\sigma)$ is standard Tikhonov, that is of optimal order, with $\beta = 1/2$ and for every $0 < \nu \leq 2$, see [61]. On the contrary, with $\beta = 1/2$ and $\nu = 2$, and by equations (5.26) and (5.27), we deduce that

$$(1 - F_{\alpha,\gamma}(\sigma))\,\sigma^2 \geq \min\{1,\gamma\}\,(1 - F_{\alpha,1}(\sigma))\,\sigma^2 \geq \frac{1}{2}\alpha, \qquad \text{for } \sigma \in [\alpha^{\frac{1}{2}}, \sigma_1].$$

Therefore, if $\|x^\dagger - x_{\alpha,r}\| = O(\alpha)$ then $x^\dagger \in \mathcal{X}_2$ by Theorem 5.7. □

A similar saturation result to Proposition 5.11 can be proved also for the fractional Tikhonov regularization.

Proposition 5.14 (Saturation for fractional Tikhonov regularization)**.** *Let* $A : \mathcal{X} \to \mathcal{Y}$ *be a compact linear operator with infinite dimensional range and let* $R_{\alpha,\gamma}$ *be the corresponding family of fractional Tikhonov regularization operators in Definition 5.12, with fixed* $\gamma \geq 1/2$. *Let* $\alpha = \alpha\left(\delta, b^\delta\right)$ *be any parameter choice rule. If*

$$\sup\left\{\|x^\delta_{\alpha,\gamma} - x^\dagger\| :\ \|Q\left(y - b^\delta\right)\| \leq \delta\right\} = o\left(\delta^{\frac{2}{3}}\right), \tag{5.28}$$

then $x^\dagger = 0$, *where we indicated with* Q *the orthogonal projector onto* $\overline{R(A)}$.

Proof. If $\gamma = 1$, the thesis follows from the saturation result for standard Tikhonov [61, Proposition 5.3]. For $\gamma \neq 1$, recalling that

$$x_{\alpha,\gamma} - x^\dagger = \sum_{\sigma_l > 0} \left(F_{\alpha,\gamma}\left(\sigma_l\right) - 1\right) \sigma_l^{-1} \langle b, u_l \rangle v_l,$$

by equations (5.26) and (5.27), we obtain

$$\|x_{\alpha,\gamma} - x^\dagger\| > c \|x_{\alpha,1} - x^\dagger\|,$$

where $c = \min\{1, \gamma\}$ and $x_{\alpha,1}$ is standard Tikhonov. Let us define

$$\phi_\gamma\left(b\right) := \|x_{\alpha,\gamma} - x^\dagger\|.$$

Then, by the continuity of ϕ_γ, there exists $\delta > 0$ such that, for every $b^\delta \in \overline{B}_\delta\left(b\right)$, we find

$$\phi_\gamma\left(b^\delta\right) > c \cdot \phi_1\left(b^\delta\right),$$

with $\overline{B}_\delta\left(b\right)$ being the closure of the ball of center b and radius δ. Passing to the $\sup$ we obtain that

$$\sup\left\{\|x^\delta_{\alpha,\gamma} - x^\dagger\| : \|Q\left(b - b^\delta\right)\| \leq \delta\right\} \geq c \cdot \sup\left\{\|x^\delta_{\alpha,1} - x^\dagger\| : \|Q\left(b - b^\delta\right)\| \leq \delta\right\}.$$

Therefore, using relation (5.28), we deduce

$$\sup\left\{\|x^\delta_{\alpha,1} - x^\dagger\| : \|b - b^\delta\| \leq \delta\right\} = o\left(\delta^{\frac{2}{3}}\right),$$

and the thesis follows again from the saturation result for standard Tikhonov, cf. [61, Proposition 5.3]. □

Differently from the weighted Tikhonov regularization, for the fractional Tikhonov method, it is not possible to overcome the saturation result of classical Tikhonov, even for a large γ.

5.3 Stationary iterated regularization

We define new iterated regularization methods based on weighed and fractional Tikhonov regularization using the same iterative refinement strategy of iterated Tikhonov regularization, see Section 2.2.2. We will show that the iterated methods go beyond the saturation results proved in the previous section. In this section the regularization parameter will still be α with the iteration step, k, assumed to be fixed. On the contrary, in Section 5.4.1, we will analyze the nonstationary counterpart of this iterative method, in which α will be replaced by a pre-fixed sequence $\{\alpha_k\}$ and we will be concerned on the rate of convergence with respect to the index k.

5.3.1 Iterated weighted Tikhonov regularization

We propose now an iterated regularization method based on weighted Tikhonov

Definition 5.15 (Stationary iterated weighted Tikhonov)**.** *We define the* stationary iterated weighted Tikhonov method *(*SIWT*) as*

$$\begin{cases} x_{\alpha,r}^0 := 0; \\ \left((A^*A)^{\frac{r+1}{2}} + \alpha I \right) x_{\alpha,r}^k := (A^*A)^{\frac{r-1}{2}} A^*b + \alpha x_{\alpha,r}^{k-1}, \end{cases} \tag{5.29}$$

with $\alpha > 0$ and $r \geq 0$, or equivalently

$$\begin{cases} x_{\alpha,r}^0 := 0 \\ x_{\alpha,r}^k := \operatorname{argmin}_{x \in \mathcal{X}} \left\{ \|Ax - b\|_W^2 + \alpha \|x - x_{\alpha,r}^{k-1}\|^2 \right\}, \end{cases}$$

where $\|\cdot\|_W$ is the semi-norm introduced in (5.20)*. We define $x_{\alpha,r}^{k,\delta}$ as the k-th iteration of weighted Tikhonov if $b = b^\delta$.*

Proposition 5.16. *For any given $k \in \mathbb{N}$ and $r > 0$, the SIWT in* (5.29) *is a filter based regularization method, with filter function*

$$F_{\alpha,r}^{(k)}(\sigma) = \frac{\left(\sigma^{r+1} + \alpha\right)^k - \alpha^k}{\left(\sigma^{r+1} + \alpha\right)^k}.$$

Moreover, the method is of optimal order, under the a-priori assumption $x^\dagger \in \mathcal{X}_{\nu,\rho}$, for $r > 0$ and $0 < \nu \leq k(r+1)$, with best convergence rate $\|x^\dagger - x_{\alpha,r}^{k,\delta}\| = O\left(\delta^{\frac{k(r+1)}{1+k(r+1)}}\right)$, that is obtained for $\alpha = \left(\frac{\delta}{\rho}\right)^{\frac{k(r+1)}{1+\nu}}$, with $\nu = k(r+1)$. On the other hand, if $\|x^\dagger - x_{\alpha,r}^k\| = O\left(\alpha^k\right)$, then $x^\dagger \in \mathcal{X}_{k(r+1)}$.

Proof. Multiplying both sides of (5.29) by $\left((A^*A)^{\frac{r+1}{2}} + \alpha I\right)^{k-1}$ and iterating the process, we get

$$\begin{aligned} \left((A^*A)^{\frac{r+1}{2}} + \alpha I\right)^k x_{\alpha,r}^k &= \left\{ \sum_{j=0}^{n-1} \alpha^j \left((A^*A)^{\frac{r+1}{2}} + \alpha I\right)^{k-1-j} \right\} (A^*A)^{\frac{r-1}{2}} A^*b \\ &= \left[\left((A^*A)^{\frac{r+1}{2}} + \alpha I\right)^k - \alpha^k I \right] (A^*A)^{-1} A^*b. \end{aligned}$$

Therefore, the filter function in (5.6) is equal to

$$F_{\alpha,r}^{(k)}(\sigma) = \frac{\left(\sigma^{r+1} + \alpha\right)^k - \alpha^k}{\left(\sigma^{r+1} + \alpha\right)^k},$$

as we stated. Condition (5.9) is straightforward to verify. Moreover, note that

$$\begin{aligned} F_{\alpha,r}^{(k)}(\sigma) &= \frac{\left(\sigma^{r+1} + \alpha\right)^k - \alpha^k}{\left(\sigma^{r+1} + \alpha\right)^k} \\ &= \frac{\sigma^{r+1}}{\sigma^{r+1} + \alpha} \cdot \frac{\left(\sum_{j=0}^{k-1} \alpha^j \left(\sigma^{r+1} + \alpha\right)^{k-1-j}\right)}{\left(\sigma^{r+1} + \alpha\right)^{k-1}} \\ &= F_{\alpha,r}(\sigma) \cdot \left(1 + \left(\frac{\alpha}{\sigma^{r+1} + \alpha}\right) + \cdots + \left(\frac{\alpha}{\sigma^{r+1} + \alpha}\right)^{k-1}\right), \end{aligned}$$

from which it follows that

$$F_{\alpha,r}(\sigma) \leq F_{\alpha,r}^{(k)}(\sigma) \leq k F_{\alpha,r}(\sigma).$$

Therefore, conditions (5.7), (5.8) and (5.14a) follows immediately by the regularity of the weighted Tikhonov filter method for $r > 0$ and by the order optimality for $r > 0$. Finally, condition (5.14b) becomes

$$\sup_{\sigma\in[0,\sigma_1]} \left| \frac{\alpha^k \sigma^\nu}{\left(\sigma^{r+1}+\alpha\right)^k} \right|,$$

and deriving one checks that it is bounded by $\alpha^{\beta\nu}$, with $\beta = 1/(r+1)$, if and only if $0 < \nu \leq k(r+1)$. Applying now Proposition 5.6 the rest of the thesis follows.

On the contrary, if we define $\beta = 1/(r+1)$ and $\nu = k(r+1)$, then we deduce that

$$\left(1 - F^{(k)}_{\alpha,r}(\sigma)\right)\sigma^\nu = \frac{\alpha^k\sigma^\nu}{\left(\sigma^{r+1}+\alpha\right)^k} \geq \frac{1}{2^k}\alpha^k \qquad \text{for } \sigma \in [\alpha^\beta, \sigma_1].$$

Therefore, if $\|x^\dagger - x^n_{\alpha,r}\| = O\left(\alpha^k\right)$, then by Theorem 5.7 it follows that $x^\dagger \in \mathcal{X}_{k(r+1)}$. □

If k is large, then we note that the convergence rate approaches $O(\delta)$ also for a fixed small r. The study of the convergence for increasing k and fixed α will be dealt with in Section 5.4.1.

5.3.2 Iterated fractional Tikhonov regularization

With the same path as in the previous subsection, we propose here the stationary iterated version of the fractional Tikhonov method.

Definition 5.17 (Stationary iterated fractional Tikhonov)**.** *We define the* stationary iterated fractional Tikhonov method *(*SIFT*) as*

$$\begin{cases} x^0_{\alpha,\gamma} := 0; \\ \left(A^*A+\alpha I\right)^\gamma x^k_{\alpha,\gamma} := \left(A^*A\right)^{\gamma-1} A^*b + \left[\left(A^*A+\alpha I\right)^\gamma - \left(A^*A\right)^\gamma\right] x^{n-1}_{\alpha,\gamma}, \end{cases} \tag{5.30}$$

with $\gamma \geq 1/2$. We define $x^{k,\delta}_{\alpha,\gamma}$ for the n-th iteration of fractional Tikhonov if $b = b^\delta$.

Proposition 5.18. *For any given $k \in \mathbb{N}$ and $\gamma \geq 1/2$, the SIFT in* (5.30) *is a filter based regularization method, with filter function*

$$F^{(k)}_{\alpha,\gamma}(\sigma) = \frac{\left(\sigma^2+\alpha\right)^{\gamma k} - \left[\left(\sigma^2+\alpha\right)^\gamma - \sigma^{2\gamma}\right]^k}{\left(\sigma^2+\alpha\right)^{\gamma k}}. \tag{5.31}$$

Moreover, the method is of optimal order, under the a-priori assumption $x^\dagger \in \mathcal{X}_{\nu,\rho}$, for $\gamma \geq 1/2$ and $0 < \nu \leq 2k$, with best convergence rate $\|x^\dagger - x^{k,\delta}_{\alpha,\gamma}\| = O\left(\delta^{\frac{2k}{2k+1}}\right)$, that is obtained for $\alpha = \left(\frac{\delta}{\rho}\right)^{\frac{2k}{\nu+1}}$, with $\nu = 2k$. On the other hand, if $\|x^\dagger - x^k_{\alpha,\gamma}\| = O\left(\alpha^k\right)$, then $x^\dagger \in \mathcal{X}_{2k}$.

Proof. Multiplying both sides of (5.31) by $\left(A^*A+\alpha I\right)^{(k-1)\gamma}$ and iterating the process, we get

$$\begin{aligned} \left(A^*A+\alpha I\right)^{k\gamma} x^k_{\alpha,\gamma} &= \left\{\sum_{j=0}^{k-1} \left(A^*A+\alpha I\right)^{j\gamma}\left[\left(A^*A+\alpha I\right)^\gamma - \left(A^*A\right)^\gamma\right]^{k-1-j}\right\}\left(A^*A\right)^{\gamma-1}A^*b \\ &= \left\{\left(A^*A+\alpha I\right)^{\gamma k} - \left[\left(A^*A+\alpha I\right)^\gamma - \left(A^*A\right)^\gamma\right]^k\right\}\left(A^*A\right)^{-1}A^*b, \end{aligned}$$

where we used the fact that $(A^*A+\alpha I)^{-\gamma}$ and $[(A^*A+\alpha I)^{\gamma}-(A^*A)^{\gamma}]$ commute. Therefore, the filter function in (5.6) is given by

$$F^k_{\alpha,\gamma}(\sigma)=\frac{\left(\sigma^2+\alpha\right)^{\gamma k}-\left[\left(\sigma^2+\alpha\right)^{\gamma}-\sigma^{2\gamma}\right]^k}{\left(\sigma^2+\alpha\right)^{\gamma k}},$$

as we stated. We observe that

$$\begin{aligned}F^{(k)}_{\alpha,\gamma}(\sigma)&=\frac{\left(\sigma^2+\alpha\right)^{\gamma k}-\left[\left(\sigma^2+\alpha\right)^{\gamma}-\sigma^{2\gamma}\right]^k}{\left(\sigma^2+\alpha\right)^{\gamma k}}\\&=\frac{\sigma^{2\gamma}}{\left(\sigma^2+\alpha\right)^{\gamma}}\cdot\frac{1}{\left(\sigma^2+\alpha\right)^{\gamma(k-1)}}\cdot\sum_{j=0}^{k-1}\left(\sigma^2+\alpha\right)^{\gamma j}\left[\left(\sigma^2+\alpha\right)^{\gamma}-\sigma^{2\gamma}\right]^{k-1-j}\\&=\frac{\sigma^{2\gamma}}{\left(\sigma^2+\alpha\right)^{\gamma}}\cdot\left\{1+\left[1-\left(\frac{\sigma^2}{\sigma^2+\alpha}\right)^{\gamma}\right]+\cdots+\left[1-\left(\frac{\sigma^2}{\sigma^2+\alpha}\right)^{\gamma}\right]^{k-1}\right\},\end{aligned}$$

from which we deduce that

$$F^{(k)}_{\alpha,\gamma}(\sigma)\le kF_{\alpha,\gamma}(\sigma).$$

Therefore, since $F_{\alpha,\gamma}$ is a regularization method of optimal order, conditions (5.7), (5.8) and (5.14a) are satisfied. Moreover, it is easy to check condition (5.9) and so we get the regularity for the method. It remains to check condition (5.14b) for the order optimality.

From equations (5.26) and (5.27) we deduce that

$$\begin{aligned}1-F^{(k)}_{\alpha,\gamma}(\sigma)&=\left[\frac{\left(\sigma^2+\alpha\right)^{\gamma}-\sigma^{2\gamma}}{\left(\sigma^2+\alpha\right)^{\gamma}}\right]^k\\&=\left[1-\frac{\sigma^{2\gamma}}{\left(\sigma^2+\alpha\right)^{\gamma}}\right]^k\\&=\left(1-F_{\alpha,\gamma}(\sigma)\right)^k\\&\le\left(\max\{1,\gamma\}\right)^k\left(1-F_{\alpha,1}(\sigma)\right)^k\\&=c\left(1-F^k_{\alpha,1}(\sigma)\right),\end{aligned}\tag{5.32}$$

where $F_{\alpha,1}(\sigma)$ is the standard Tikhonov filter and $F^{(k)}_{\alpha,1}(\sigma)$ is the filter function of the stationary iterated Tikhonov, i.e., $F^{(k)}_{\alpha,1}(\sigma)=\frac{\left(\sigma^2+\alpha\right)^k-\alpha^k}{\left(\sigma^2+\alpha\right)^k}$. Now condition (5.14b) follows from the properties of stationary iterated Tikhonov, with $\beta=1/2$ and $0<\nu\le 2k$, see [80, p. 124]. By applying Proposition 5.6 we get the best convergence rate, $O\left(\delta^{\frac{2k}{2k+1}}\right)$.

On the contrary, set $\beta=1/2$ and $\nu=2k$. First, let us observe that from equations (5.32) and (5.26), (5.27), we infer that

$$1-F^{(k)}_{\alpha,\gamma}(\sigma)\ge\left(\min\{1,\gamma\}\right)^k\left(1-F^{(k)}_{\alpha,1}(\sigma)\right).$$

Then, we deduce that

$$\begin{aligned}\left(1-F^{(k)}_{\alpha,\gamma}(\sigma)\right)\sigma^{\nu}&\ge c\frac{\alpha^k\sigma^{2k}}{\left(\sigma^2+\alpha\right)^k}\\&\ge c\alpha^k\qquad\text{for }\sigma\in[\alpha^{\beta},\sigma_1].\end{aligned}$$

Therefore, if $\|x^\dagger - x^k_{\alpha,\gamma}\| = O\left(\alpha^k\right)$, then $x^\dagger \in \mathcal{X}_{2n}$ by Theorem 5.7. □

The previous proposition shows that, similarly to SIWT, a large k allows to overcome the saturation result in Proposition 5.14. The study of the convergence for increasing k and fixed α will be dealt with in Section 5.4.2.

5.4 Nonstationary iterated regularization

5.4.1 Nonstationary iterated weighted Tikhonov regularization

We introduce a nonstationary version of the iteration (5.29). We study the convergence and we prove that the new iteration is a regularization method.

Definition 5.19. *Let $\{\alpha_k\}_{k\in\mathbb{N}}, \{r_k\}_{k\in\mathbb{N}} \subset \mathbb{R}_{>0}$ be sequences of positive real numbers. We define a nonstationary iterated weighted Tikhonov method (NSIWT) as follows*

$$\begin{cases} x^0_{\alpha_0,r_0} := 0, \\ \left[(A^*A)^{\frac{r_k+1}{2}} + \alpha_k I\right] x^k_{\alpha_k,r_k} := (A^*A)^{\frac{r_k-1}{2}} A^*b + \alpha_k x^{k-1}_{\alpha_{k-1},r_{k-1}}, \end{cases} \tag{5.33}$$

or equivalently

$$\begin{cases} x^0_{\alpha_0,r_0} := 0, \\ x^k_{\alpha_k,r_k} := \operatorname{argmin}_{x\in\mathcal{X}} \left\{\|Ax-b\|^2_{W_k} + \alpha_k \|x - x^{k-1}_{\alpha_{k-1},r_{k-1}}\|\right\}^2, \end{cases} \tag{5.34}$$

where $\|\cdot\|_{W_k}$ is the semi-norm introduced by the operator $W_k := (AA^*)^{\frac{r_k-1}{2}}$ and depending on k, due to the nonstationary character of r_k.

Convergence analysis

We are concerned about the properties of the sequence $\{\alpha_k\}$ such that the iteration (5.33) shall converge. To this aim we need some preliminary lemmas.

Remark 5.20. *Hereafter, without loss of generality, we will assume that $\sigma_1 = 1$, namely $\|A\| = 1$.*

Lemma 5.21. *Let $\{t_k\}_{k\in\mathbb{N}}$ be a sequence of real numbers such that $0 \le t_k < 1$ for every n. Then*

$$\prod_{k=1}^{\infty} (1-t_k) > 0 \qquad \text{if and only if} \qquad \sum_{k=1}^{\infty} t_k < \infty.$$

Proof. See [117, Theorem 15.5] □

Lemma 5.22. *Let $\{t_j\}_{j\in\mathbb{N}}$ be a sequence of positive real numbers and let $N > 0$. Then*

$$\sum_{j=1}^{k} t_j \sim c \sum_{j=N}^{k} t_j,$$

with $c > 0$ a constant independent of N and k (in particular, $c = 1$ when $\sum_{j=N}^{\infty} t_j = \sum_{j=1}^{\infty} t_j = \infty$).

Proof. Obviously, both the series converge or diverge simultaneously due to the Asymptotic Comparison test. If they converge, the thesis follows trivially. On the contrary, if they both

diverge then we conclude by observing that $\sum_{j=N}^{k} t_j / \sum_{j=1}^{k} t_j$ is a monotonically increasing sequence bounded from above by 1. Indeed, if we set

$$A_k := \sum_{j=N}^{k} t_j, \qquad B_k := \sum_{j=1}^{k} t_j,$$

for every $j \geq N$ and for every $x \geq 0$ the function

$$h_k(x) = \frac{A_k + x}{B_k + x}$$

is monotone increasing with $h_k(x) \leq 1$. Then $A_{k+1}/B_{k+1} \geq A_k/B_k$ for every k and it is easy to see that $\sup_k \{A_k/B_k\} = 1$. □

Lemma 5.23. *For every sequence $\{t_k\}_{k\in\mathbb{N}} \subset (0,\infty)$ such that $\lim_{k\to\infty} t_k = t \in (0,\infty]$, we find*

$$\sum_{k=1}^{k} \frac{1}{t_k} \sim c \sum_{k=1}^{k} \frac{1}{1+t_k}, \qquad c > 0,$$

where $\sim$ denotes the asymptotic equivalence.

Proof. If $\lim_{j\to\infty} t_j = t \in (0,\infty]$, then

$$\frac{1}{t_j} \sim \left(1 + \frac{1}{t}\right) \frac{1}{1+t_j}, \tag{5.35}$$

where $1/t = 0$ if $t = \infty$. Therefore, from the Asymptotic Comparison test for series, both series converge or diverge simultaneously. When they converge the thesis follows trivially. If we set

$$X_k := \frac{\sum_{j=1}^{k} \frac{1}{t_j}}{\sum_{j=1}^{k} \frac{1}{1+t_j}},$$

we want to show that the limit of X_k exists finite and, moreover, that $\lim_{k\to\infty} X_k = 1 + 1/t$. Indeed, for any fixed $\epsilon > 0$ there exists K_ϵ^1 such that for any $j \geq K_\epsilon^1$ it holds that

$$\frac{1}{t_j} < \left(1 + \frac{1}{t} + \frac{\epsilon}{2}\right) \frac{1}{1+t_j}, \tag{5.36}$$

and for any fixed ϵ and K_ϵ^1, there exists K_ϵ^2 such that for every $n \geq K_\epsilon^2$ it holds that

$$\frac{\sum_{j=1}^{K_\epsilon^1} \frac{1}{t_j}}{\sum_{j=1}^{k} \frac{1}{1+t_j}} < \frac{\epsilon}{2}. \tag{5.37}$$

Hence, for any $n \geq \max\{K_\epsilon^1, K_\epsilon^2\}$, thanks to (5.36) and (5.37), we have that

$$X_k = \frac{\sum_{j=1}^{k} \frac{1}{t_j}}{\sum_{j=1}^{k} \frac{1}{1+t_j}} < \frac{\sum_{j=1}^{K_\epsilon^1} \frac{1}{t_j}}{\sum_{j=1}^{k} \frac{1}{1+t_j}} + \left(1 + \frac{1}{t} + \frac{\epsilon}{2}\right) \frac{\sum_{j=K_\epsilon^1+1}^{k} \frac{1}{1+t_j}}{\sum_{j=1}^{k} \frac{1}{1+t_j}} < \frac{\epsilon}{2} + 1 + \frac{1}{t} + \frac{\epsilon}{2} = 1 + \frac{1}{t} + \epsilon.$$

On the other hand, there exists K_ϵ^3 such that for every $k \geq K_\epsilon^3$ it holds

$$\frac{1}{t_j} > \left(1 + \frac{1}{t} - \frac{\epsilon}{2}\right) \frac{1}{1+t_j}, \tag{5.38}$$

and, by Lemma 5.22, for any fixed K_ϵ^3 and for any fixed $\delta < \frac{\epsilon}{2}\left(1+\frac{1}{t}-\frac{\epsilon}{2}\right)^{-1}$, there exists K_ϵ^4 such that for every $n \geq K_\epsilon^4$ it holds

$$\frac{\sum_{k=K_\epsilon^3+1}^{k} \frac{1}{1+t_j}}{\sum_{j=1}^{k} \frac{1}{1+t_j}} > (1-\delta)\,. \tag{5.39}$$

Hence, fo any $k \geq \max\{K_\epsilon^3, K_\epsilon^4\}$, thanks to (5.38) and (5.39), we have that

$$\begin{aligned} X_k =& \frac{\sum_{j=1}^{k} \frac{1}{t_j}}{\sum_{j=1}^{k} \frac{1}{1+t_j}} \\ &> \frac{\sum_{j=1}^{K_\epsilon^1} \frac{1}{t_j}}{\sum_{j=1}^{k} \frac{1}{1+t_j}} + \left(1+\frac{1}{t}-\frac{\epsilon}{2}\right)\frac{\sum_{j=K_\epsilon^1+1}^{k} \frac{1}{1+t_j}}{\sum_{j=1}^{k} \frac{1}{1+t_j}} \\ &> \left(1+\frac{1}{t}-\frac{\epsilon}{2}\right)(1-\delta) \\ &> 1+\frac{1}{t}-\epsilon. \end{aligned}$$

Then, choosing $k \geq \max\{K_\epsilon^i : i = 1,2,3,4\}$, the proof is concluded. □

We can now prove a necessary and sufficient condition for the sequence $\{\alpha_k\}$ to have the convergence of NSIWT.

Theorem 5.24. *For every $x^\dagger \in \mathcal{X}$, the* NSIWT *method* (5.33) *converges to $x^\dagger \in \mathcal{X}$ as $k \to \infty$ if and only if $\sum_{j=1}^{k} \frac{\sigma^{r_j+1}}{\sigma^{r_j+1}+\alpha_j}$ diverges for every $\sigma > 0$.*

Proof. Rewriting equation (5.33) and reminding that $b = Ax^\dagger$, we have

$$\begin{aligned} x_{\alpha_k,r_k}^k &= \left[(A^*A)^{\frac{r_k+1}{2}} + \alpha_k I\right]^{-1}(A^*A)^{\frac{r_k+1}{2}}x^\dagger + \alpha_k\left[(A^*A)^{\frac{r_k+1}{2}} + \alpha_k I\right]^{-1} x_{\alpha_{k-1},r_{k-1}}^{k-1} \\ &= \left\{I - \alpha_k\left[(A^*A)^{\frac{r_k+1}{2}} + \alpha_k I\right]^{-1}\right\}x^\dagger + \alpha_k\left[(A^*A)^{\frac{r_k+1}{2}} + \alpha_k I\right]^{-1} x_{\alpha_{k-1},r_{k-1}}^{k-1}, \end{aligned}$$

from which it follows that

$$\begin{aligned} x^\dagger - x_{\alpha_k,r_k}^k &= \alpha_k\left[(A^*A)^{\frac{r_k+1}{2}} + \alpha_k I\right]^{-1}\left(x^\dagger - x_{\alpha_{k-1},r_{k-1}}^{k-1}\right) \\ &= (\cdots) \text{ iterating the process } k-1 \text{ times} \\ &= \prod_{j=1}^{k} \alpha_j \left[(A^*A)^{\frac{r_j+1}{2}} + \alpha_j I\right]^{-1} x^\dagger \end{aligned} \tag{5.40}$$

since $x_{\alpha_0,r_0}^0 := 0$. As a consequence, the method shall converge if and only if

$$\lim_{k\to\infty} \left\| \prod_{j=1}^{k} \alpha_j \left[(A^*A)^{\frac{r_j+1}{2}} + \alpha_j I\right]^{-1} x^\dagger \right\| = 0$$

for every $x^\dagger \in \mathcal{X}$, namely, if and only if

$$\lim_{k\to\infty} \int_{\lambda(A^*A)} \left| \prod_{j=1}^{k} \frac{\alpha_k}{\sigma^{r_j+1}+\alpha_j} \right|^2 \mathrm{d}\langle E_{\sigma^2} x^\dagger, x^\dagger\rangle = 0$$

for every Borel-measure $\langle Ex^\dagger, x^\dagger\rangle$ induced by $x^\dagger \in \mathcal{X}$. Since

$$\left| \prod_{j=1}^{k} \frac{\alpha_j}{\sigma^{r_j+1}+\alpha_j} \right|^2 \leq 1$$

for every k, and since

$$\int_{\lambda(A^*A)} \mathrm{d}\langle E_{\sigma^2} x^\dagger, x^\dagger\rangle = \|x^\dagger\|^2,$$

the Dominated Convergence Theorem [117, Theorem 1.34, pag. 26] implies

$$\begin{aligned}
&\lim_{k\to\infty} \int_{\lambda(A^*A)} \left| \prod_{j=1}^{k} \frac{\alpha_j}{\sigma^{r_j+1}+\alpha_j} \right|^2 \mathrm{d}\langle E_{\sigma^2} x^\dagger, x^\dagger\rangle \\
&= \int_{\lambda(A^*A)} \lim_{k\to\infty} \left| \prod_{j=1}^{k} \frac{\alpha_j}{\sigma^{r_j+1}+\alpha_j} \right|^2 \mathrm{d}\langle E_{\sigma^2} x^\dagger, x^\dagger\rangle.
\end{aligned}$$

Hence, the NSIWT method is convergent if and only if

$$\prod_{j=1}^{\infty} \frac{\alpha_j}{\sigma^{r_j+1}+\alpha_k} = \prod_{j=1}^{\infty} \left(1 - \frac{\sigma^{r_j+1}}{\sigma^{r_j+1}+\alpha_k}\right) = 0,$$

for $\langle Ex^\dagger, x^\dagger\rangle$-a.e. σ^2, i.e., for every $\sigma \in \sigma(A) \setminus \{0\}$. Applying now Lemma 5.21 the thesis follows. □

Corollary 5.25. *(1) If $\sup_{j\in\mathbb{N}}\{r_j\} = r \in [0,\infty)$, then the NSIWT method converges if and only if $\sum_{j=1}^{k} \alpha_j^{-1}$ diverges.*

(2) Let $\lim_{j\to\infty} r_j = \infty$ monotonically and let us set $\beta_k = \sum_{j=1}^{k} \alpha_j^{-1}$. If $\lim_{k\to\infty} \beta_k^{1/r_k} = \infty$, then the NSIWT method converges.

Proof. (1) For every $\sigma \in \sigma(A) \setminus \{0\}$, we observe that

$$\sum_{j=1}^{\infty} \frac{\sigma^{r+1}}{\sigma^{r+1}+\alpha_j} \leq \sum_{j=1}^{\infty} \frac{\sigma^{r_j+1}}{\sigma^{r_j+1}+\alpha_j} \leq \sum_{j=1}^{\infty} \frac{1}{1+\alpha_j} \leq \sum_{j=1}^{\infty} \frac{1}{\alpha_j}. \tag{5.41}$$

If the NSIWT method converges then, by Theorem 5.24 and by (5.41), $\sum_{j=1}^{\infty} \frac{\sigma^{r_j+1}}{\sigma^{r_j+1}+\alpha_j}$ diverges and hence $\sum_{j=1}^{\infty} \frac{1}{\alpha_j} = \infty$. On the other hand, if $\sum_{j=1}^{\infty} \alpha_j^{-1} = \infty$, then we can possibly have three different cases: $\lim_{j\to\infty} \alpha_j \in [0,\infty)$, $\nexists \lim_{j\to\infty} \alpha_j$ or $\lim_{j\to\infty} \alpha_j = \infty$. In the first two cases, $\frac{\sigma^{r+1}}{\sigma^{r+1}+\alpha_j} \not\to 0$ for every $\sigma > 0$, and then the corresponding series diverges. In the latter case instead $\alpha_j^{-1} \sim c_{\sigma,r} \frac{\sigma^{r+1}}{\sigma^{r+1}+\alpha_j}$ for every $\sigma > 0$, and hence the series $\sum_{j=1}^{k} \alpha_j^{-1}$ and $\sum_{j=1}^{k} \frac{\sigma^{r+1}}{\sigma^{r+1}+\alpha_j}$ converge or diverge simultaneously by the Asymptotic Comparison test.

Then, by $\sum_{j=1}^{\infty} \alpha_j^{-1} = \infty$, we deduce that $\sum_{j=1}^{\infty} \frac{\sigma^{r_j+1}}{\sigma^{r_j+1}+\alpha_k}$ diverges for every $\sigma > 0$ and the NSIWT method converges.

(2) Note that

$$\lim_{k\to\infty} \beta_k^{1/r_k} = \infty \qquad \Longleftrightarrow \qquad \lim_{k\to\infty} \sigma^{r_k} \left(\sum_{j=1}^{k} \alpha_j^{-1} \right) = \infty \quad \forall \sigma \in \sigma(A) \setminus \{0\},$$

namely,

$$\lim_{k\to\infty} \beta_k^{1/r_k} = \infty \qquad \Longleftrightarrow \qquad \left(\sum_{j=1}^{k} \alpha_j^{-1} \right)^{-1} = o(\sigma^{r_j}) \quad \forall \sigma \in \sigma(A) \setminus \{0\}. \tag{5.42}$$

In fact, it holds

($\Rightarrow$) We have that

$$\sigma^{r_k} \beta_k = \left(\sigma \beta_k^{1/r_k} \right)^{r_k}.$$

Since, by hypothesis, $\lim \beta_k^{1/r_k} = \infty$, then

$$\left(\sigma \beta_k^{1/r_k} \right)^{r_k} \to \infty \qquad \text{as } n \to \infty.$$

Indeed, ∞^{∞} is not an indeterminate form.

($\Leftarrow$) By contradiction, let us suppose that $\beta_k^{1/r_k} \nrightarrow \infty$. Since β_k is a monotone increasing sequence, by monotonicity it admits limit, and it follows that

$$\lim_k \beta_k^{1/r_k} = c \in (0, \infty).$$

Then,there exists $\hat{\sigma} \in (0,1)$ such that $\hat{\sigma} c \in (0,1)$. Therefore

$$\lim_{k\to\infty} \hat{\sigma}^{r_k} \beta_k = \lim_{n\to\infty} \left(\hat{\sigma} \beta_k^{1/r_k} \right)^{r_k} = \lim_{n\to\infty} (\hat{\sigma} c)^{r_k} = 0,$$

a contradiction, since by hypothesis

$$\sigma^{r_k} \beta_k \to \infty \quad \forall\, \sigma \in [0,1].$$

We can assume that $0 < \sigma < 1$. For $\sigma = 1$ the result is indeed trivial owing to the equivalence

$$\sum_{j=1}^{\infty} \frac{1}{1+\alpha_j} = \infty \Longleftrightarrow \sum_{j=1}^{\infty} \alpha_j^{-1} = \infty \qquad \text{(see the previous point).}$$

Let us fix $\sigma \in (0,1)$ and for the sake of simplicity let suppose that $\{\alpha_j\}$ admits limit, i.e., $\lim_{j\to\infty} \alpha_j \in [0, \infty]$. We have two cases:

$$\lim_{j\to\infty} \frac{\alpha_j}{\sigma^{r_j+1}} = 0 \quad \text{or} \quad \lim_{j\to\infty} \frac{\alpha_j}{\sigma^{r_j+1}} \in (0, \infty].$$

In the first case, $\frac{\sigma^{r_j+1}}{\sigma^{r_j+1}+\alpha_j} \nrightarrow 0$ for $j \to \infty$, then the corresponding series $\sum_{j=1}^{k} \frac{\sigma^{r_j+1}}{\sigma^{r_j+1}+\alpha_j}$ diverges. In this case we did not use (5.42), but note that

$$\sigma^{k+1}\alpha_k^{-1} \leq \sigma^{k+1}\sum_{j=1}^{k}\alpha_j^{-1}$$

and then, if $\lim_{j\to\infty} \alpha_j/\sigma^{r_j+1} = 0$, it holds $\left(\sum_{j=1}^{k}\alpha_j^{-1}\right)^{-1} = o(\sigma^{r_k+1})$. In the second case, we have $\frac{1}{\sigma^{r_j+1}+\alpha_j} \sim c\alpha_j^{-1}$, $c > 0$, for $j \to \infty$. Therefore, there exists $K = K(\sigma)$ such that $\frac{1}{\sigma^{r_j+1}+\alpha_j} \geq \frac{c}{2}\alpha_j^{-1}$ for every $j \geq K$. Hence, fixed $k > K$, we have

$$\begin{aligned}\frac{c}{2}\sigma^{r_k+1}\sum_{j=K}^{k}\alpha_j^{-1} &\leq \sigma^{r_k+1}\left(\sum_{j=1}^{K-1}\frac{1}{\sigma^{r_j+1}+\alpha_j} + \frac{c}{2}\sum_{j=K}^{k}\alpha_j^{-1}\right)\\ &\leq \sum_{j=1}^{k}\frac{\sigma^{r_k+1}}{\sigma^{r_j+1}+\alpha_j}\\ &\leq \sum_{j=1}^{k}\frac{\sigma^{r_j+1}}{\sigma^{r_j+1}+\alpha_j},\end{aligned}$$

where the last inequality stands in virtue of the monotonicity of $\{r_j\}$.

Since, by Lemma 5.22, $\sum_{j=K}^{k}\alpha_j^{-1} \sim c\sum_{j=1}^{k}\alpha_j^{-1}$ then, by the preceding inequalities, the hypothesis $\left(\sum_{j=1}^{k}\alpha_j^{-1}\right)^{-1} = o(\sigma^{r_k+1})$ implies that $\sum_{j=1}^{k}\frac{\sigma^{r_j+1}}{\sigma^{r_j+1}+\alpha_j} = \infty$. Finally, due to the arbitrarily choice of σ, we can conclude that $\sum_{j=1}^{k}\frac{\sigma^{r_j+1}}{\sigma^{r_j+1}+\alpha_j}$ diverges for every $\sigma \in \sigma(A)\setminus\{0\}$, and therefore the NSIWT method converges. If $\{\alpha_j\}$ does not have limit, then the proof can be carried out identically but handling with more care the different cases

$$\liminf_{j\to\infty}\frac{\alpha_j}{\sigma^{r_j+1}} = 0 \quad \text{or} \quad \liminf_{j\to\infty}\frac{\alpha_j}{\sigma^{r_j+1}} \in (0,\infty].$$

□

Corollary 5.25 applies immediately to the stationary case, where $\alpha_j = \alpha$ and $r_j = r$ for every $j \in \mathbb{N}$, showing that SIWT converges. On the other hand, from point *(2)* of Corollary 5.25, given a monotone divergent sequence $r_j \to \infty$ we need a sequence $\alpha_j \to 0$ such that $\alpha_j = o\left(\sigma^{r_j+1}\right)$ for every $\sigma > 0$ in order to preserve the convergence of NSIWT.

Now, we investigate the convergence rate of NSIWT.

Theorem 5.26. *Let $\{x^k_{\alpha_k,r_k}\}_{k\in\mathbb{N}}$ be a convergent sequence of the NSIWT method, with $x^\dagger \in \mathcal{X}_\nu$ for some $\nu > 0$, and let $\{\vartheta_k\}_{k\in\mathbb{N}}$ be a divergent sequence of positive real numbers. If*

$$\lim_{k\to\infty}\vartheta_k\sigma^\nu\prod_{j=1}^{k}\left(1-\frac{\sigma^{r_j+1}}{\sigma^{r_j+1}+\alpha_j}\right) = 0 \qquad \textit{for every } \sigma \in \sigma(A)\setminus\{0\}; \tag{5.43a}$$

$$\sup_{\sigma\in\sigma(A)\setminus\{0\}}\vartheta_k\sigma^\nu\prod_{j=1}^{k}\left(1-\frac{\sigma^{r_j+1}}{\sigma^{r_j+1}+\alpha_j}\right) \leq c < \infty \qquad \textit{uniformly with respect to } k, \tag{5.43b}$$

then

$$\|x^\dagger - x^k_{\alpha_k,r_k}\| = o\left(\vartheta_k^{-1}\right).$$

Proof. From equation (5.40), for $x^\dagger \in \mathcal{X}_\nu$, we have

$$\begin{aligned}\lim_{k\to\infty} \vartheta_k \|x^\dagger - x^k_{\alpha_k,r_k}\| &= \lim_{k\to\infty} \left[\int_{\lambda(A^*A)} \left|\vartheta_k \sigma^\nu \prod_{j=1}^k \left(1 - \frac{\sigma^{r_j+1}}{\sigma^{r_j+1}+\alpha_j}\right)\right|^2 \mathrm{d}\langle E_{\sigma^2}\omega,\omega\rangle\right]^{1/2} \\ &= \left[\int_{\lambda(A^*A)} \left|\lim_{k\to\infty} \vartheta_k \sigma^\nu \prod_{j=1}^k \left(1 - \frac{\sigma^{r_j+1}}{\sigma^{r_j+1}+\alpha_j}\right)\right|^2 \mathrm{d}\langle E_{\sigma^2}\omega,\omega\rangle\right]^{1/2},\end{aligned}$$

by (5.43b) and the Dominated Convergence Theorem. Now, from hypothesis (5.43a), the thesis follows. □

Corollary 5.27. *We define*

$$\beta_k = \sum_{j=1}^k \alpha_j^{-1}, \qquad \tilde{\beta}_k = \sum_{j=1}^k \frac{1}{1+\alpha_j}.$$

Let $\{r_j\}_{j\in\mathbb{N}}$ be a sequence of positive real numbers, and let $x^\dagger \in X_\nu$ for some $\nu > 0$. If

(i.1) $\sup_{j\in\mathbb{N}}\{r_j\} = r \in (0,\infty)$,

(i.2) $\lim_{k\to\infty} \beta_k = \infty$,

then

$$\|x^\dagger - x^k_{\alpha_k,r_k}\| = \begin{cases} o\left(\beta_k^{-\frac{\nu}{r+1}}\right) & \text{if } \lim_{k\to\infty} \alpha_k = \alpha \in (0,\infty] \qquad (5.44a) \\ O\left(\beta_k^{-\frac{\nu}{r+1}}\right) & \text{if } \lim_{k\to\infty} \alpha_k = 0 \text{ and } \alpha_k^{-1} \le c\beta_{k-1},\ c>0 \qquad (5.44b) \\ o\left(\tilde{\beta}_k^{-\frac{\nu}{r+1}}\right) & \text{otherwise.} \qquad (5.44c) \end{cases}$$

Proof. For the sake of simplicity, let us assume that the sequences $\{\alpha_j\}$, $\{r_j\}$ admit limits. First, note that from (i.1), (i.2) and Corollary 5.25 it follows that the NSIWT method is convergent. Now, since $1 - x \le \mathrm{e}^{-x} \le c_{\nu,r} x^{-\nu/r+1}$, and using (i.2), we have

$$\begin{aligned}\sigma^\nu \prod_{j=1}^k \left(1 - \frac{\sigma^{r_j+1}}{\sigma^{r_j+1}+\alpha_j}\right) &\le \sigma^\nu \mathrm{e}^{-\sum_{j=1}^k \frac{\sigma^{r_j+1}}{\sigma^{r_j+1}+\alpha_j}} \\ &\le \sigma^\nu \mathrm{e}^{-\sigma^{r+1}\sum_{j=1}^k \frac{1}{\sigma^{r+1}+\alpha_j}} \\ &\le c_{\nu,r}\sigma^\nu \left(\frac{1}{\sigma^{r+1}\sum_{j=1}^k \frac{1}{\sigma^{r+1}+\alpha_j}}\right)^{\frac{\nu}{r+1}} \\ &\le c_{\nu,r}\left(\sum_{j=1}^k \frac{1}{1+\alpha_j}\right)^{-\frac{\nu}{r+1}}.\end{aligned}$$

Moreover, note that $\frac{1}{1+\alpha_j} \sim c\frac{1}{1+\alpha_j/\sigma^{r_j+1}}$. Therefore, conditions (5.43a) and (5.43b) of Theorem 5.26 are satisfied with

$$\vartheta_k = \left(\sum_{j=1}^{k} \frac{1}{1+\alpha_j}\right)^{\frac{\nu}{r+1}},$$

indeed

$$\sup_{\sigma\in[0,1]} \left\{ \sigma^\nu \left(\sum_{j=1}^{k} \frac{1}{1+\alpha_j}\right)^{\frac{\nu}{r+1}} \prod_{j=1}^{k} \left(1 - \frac{\sigma^{r_j+1}}{\sigma^{r_j+1}+\alpha_j}\right) \right\} \leq c_{\nu,r},$$

and

$$\begin{aligned}
&\sigma^\nu \left(\sum_{j=1}^{k} \frac{1}{1+\alpha_j}\right)^{\frac{\nu}{r+1}} \prod_{j=1}^{k} \left(1 - \frac{\sigma^{r_j+1}}{\sigma^{r_j+1}+\alpha_j}\right) \\
&\leq \left(\sum_{j=1}^{k} \frac{1}{1+\alpha_j}\right)^{\frac{\nu}{r+1}} \mathrm{e}^{-\sum_{j=1}^{k} \frac{\sigma^{r_j+1}}{\sigma^{r_j+1}+\alpha_j}} \\
&= \left(\sum_{j=1}^{k} \frac{1}{1+\alpha_j}\right)^{\frac{\nu}{r+1}} \mathrm{e}^{-\sum_{j=1}^{k} \frac{1}{1+\alpha_j/\sigma^{r_j+1}}} \\
&\leq c \left(\sum_{j=K(\sigma)}^{k} \frac{1}{1+\alpha_j/\sigma^{r_j+1}}\right)^{\frac{\nu}{r+1}} \mathrm{e}^{-\sum_{j=K(\sigma)}^{k} \frac{1}{1+\alpha_j/\sigma^{r_j+1}}},
\end{aligned}$$

where $K(\sigma)$ is chosen such that $\frac{1}{1+\alpha_j} \leq \frac{c/2}{1+\alpha_j/\sigma^{r_j+1}}$ for every $j \geq K(\sigma)$, and the right hand side of the last inequality tends to 0 as $k \to \infty$ for every fixed σ. If $\lim_{j\to\infty} \alpha_j = \alpha \in (0,\infty]$, then $\beta_k \sim c\sum_{j=1}^{k} \frac{1}{1+\alpha_j}$ for $k \to \infty$ by Lemma 5.23. Equations (5.44a) and (5.44c) follow. Eventually, observing that $1 - \frac{\sigma^{r_j+1}}{\sigma^{r_j+1}+\alpha_j} \leq 1 - \frac{\sigma^{r+1}}{\sigma^{r+1}+\alpha_j}$, equation (5.44b) follows instead by a straightforward application of [79, Lemma 1,2,3 and Theorem 1].

In the general case where no assumptions are made on the existence of the limits for the sequences $\{\alpha_j\}$ and $\{r_j\}$, we can apply the same arguments being careful to study the $\liminf$ and $\limsup$ of these sequences. □

When $r = 1$ (classical iterated Tikhonov), equation (5.44b) is shown in [79, Theorem 1]. On the other hand, if $\lim_{n\to\infty} \alpha_k = \alpha \in (0,\infty]$, then the convergence rate is improved by the small "o".

Remark 5.28. *As we stated in* (5.44b)*, when* $\lim_{k\to\infty} \alpha_k = 0$*, to obtain a convergence rate of order* $O\left(\beta_k^{-\nu/(r+1)}\right)$ *the sequence* $\{\alpha_k\}$ *has to satisfy the condition* $\alpha_k^{-1} \leq c\beta_{k-1}$ *for a positive real number* $c > 0$*. Then,* $\sum_{j=1}^{k} \alpha_j^{-1} = \beta_k = O\left(q^k\right)$*, where* $q = (1+c) > 1$*. To overcome this bound, in virtue of Corollary 5.27, choosing sequences* $\{\hat{r}_k\}$ *and* $\{\hat{\alpha}_k\}$ *such that* $\hat{r}_k$ *diverges monotonically and* $\left(\sum_{j=1}^{k} \hat{\alpha}_j^{-1}\right)^{-1} = o\left(\sigma^{\hat{r}_k+1}\right)$ *for every* $0 < \sigma \leq 1$*, we are able to obtain a faster convergence rate, in a sense that has still to be defined. In the following Proposition 5.29 we will give the proof for a specific case.*

Following the same approach in [25, (2.3), (2.4) pag.26], we say that the sequence $\{\hat{x}_k\}$ converges uniformly faster than the sequence $\{x_k\}$ if

$$x^\dagger - \hat{x}_k = R_k \left(x^\dagger - x_k \right), \tag{5.45}$$

where $\{R_k\}$ is a sequence of operators such that $\|R_k\| \to 0$ as $k \to \infty$. We say instead that $\{\hat{x}_k\}$ converges non-uniformly faster than $\{x_k\}$ if (5.45) holds and

$$\inf_{k\in\mathbb{N}} \|R_k\| > 0, \qquad \lim_{k\to\infty} \|R_k x\| = 0 \ \text{ for every } x \in \mathcal{X}.$$

We are ready to state the following comparison result.

Proposition 5.29. *Let $\{x^k_{\alpha_k}\}$ be the sequence generated by the nonstationary iterated Tikhonov with $\alpha_k = \alpha_0 q^k$, where $\alpha_0 \in (0,\infty)$, $q \in (0,1)$, and let $\{x^k_{\hat{\alpha}_k,\hat{r}_k}\}$ be the sequence generated by NSIWT, where $\hat{\alpha}_k = 1/k!$ and $\hat{r}_k = k$, both applied to the same compact operator $A : \mathcal{X} \to \mathcal{Y}$. Then, $\{x^k_{\hat{\alpha}_k,\hat{r}_k}\}$ converges, non uniformly, faster than $\{x^k_{\alpha_k}\}$.*

Proof. Observe that the sequence $\{x^k_{\alpha_k}\}$ corresponds to a NSIWT method $\{x^k_{\alpha_k,r_k}\}$ with $r_k = 1$ for every k. Moreover, both the sequences $\{x^k_{\alpha_k}\}$ and $\{x^k_{\hat{\alpha}_k,\hat{r}_k}\}$ converge, indeed they satisfy conditions (1) and (2) of Corollary 5.25, respectively. Assuming that $x_0 = 0$ and applying the same strategy used in Theorem 5.24, without any effort it is possible to show that

$$x^\dagger - x^k_{\hat{\alpha}_k,\hat{r}_k} = \prod_{j=1}^{k} \hat{\alpha}_j \left((A^*A)^{\frac{\hat{r}_j+1}{2}} + \hat{\alpha}_j I \right)^{-1} x^\dagger,$$

$$x^\dagger = \prod_{j=1}^{k} \alpha_j^{-1} \left(A^*A + \alpha_j I \right) \left(x^\dagger - x^k_{\alpha_k} \right).$$

Therefore we find

$$\begin{aligned} x^\dagger - x^k_{\hat{\alpha}_k,\hat{r}_k} &= \left[\prod_{j=1}^{k} \hat{\alpha}_j \alpha_j^{-1} \left((A^*A)^{\frac{\hat{r}_j+1}{2}} + \hat{\alpha}_j I \right)^{-1} \left(A^*A + \alpha_j I \right) \right] \left(x^\dagger - x^k_{\alpha_k} \right) \\ &= R_k \left(x^\dagger - x^k_{\alpha_k} \right). \end{aligned}$$

Since $0 \in \lambda(A^*A)$, we infer $\|R_k\| > 1$ for every k, and hence $\inf_{k\in\mathbb{N}} \|R_k\| \geq 1$. If we prove that

$$\lim_{k\to\infty} \|R_k x\| = 0,$$

for every $x \in \mathcal{X}$, then the thesis follows. Since

$$\lim_{k\to\infty} \|R_k x\| = 0 \iff \lim_{k\to\infty} \prod_{j=1}^{k} \frac{\hat{\alpha}_j \left(\sigma^2 + \alpha_j\right)}{\alpha_j \left(\sigma^{\hat{r}_j+1} + \hat{\alpha}_j\right)} = 0 \iff \sum_{j=1}^{\infty} \frac{\alpha_j \sigma^{\hat{r}_j+1} - \hat{\alpha}_j \sigma^2}{\alpha_j \sigma^{\hat{r}_j+1} + \alpha_j \hat{\alpha}_j} = \infty \ \ \forall \sigma > 0,$$

if we substitute the values $\alpha_k = \alpha_0 q^k$, then $\hat{\alpha}_k = 1/k!$ and $\hat{r}_k = k$, we obtain

$$\sum_{j=1}^{\infty} \frac{\alpha_j \sigma^{\hat{r}_j+1} - \hat{\alpha}_j \sigma^2}{\alpha_j \sigma^{\hat{r}_j+1} + \alpha_j \hat{\alpha}_j} = \sum_{j=1}^{\infty} \frac{1 - \frac{\sigma}{\alpha_0 k! (q\sigma)^k}}{1 + \frac{1/k!}{\sigma^{k+1}}},$$

and the right hand side of the above equality diverges: indeed

$$\frac{1-\frac{\sigma}{\alpha_0 k!(q\sigma)^k}}{1+\frac{1/k!}{\sigma^{k+1}}} \longrightarrow 1 \quad \text{for every fixed } q,\sigma \in (0,1) \text{ and } \alpha_0 \in (0,\infty)\,.$$

□

Analysis of convergence for perturbed data

Let us now consider $b^\delta = b + \delta\eta$, with $b \in R(A)$ and $\|\eta\| = 1$, i.e., $\|b^\delta - b\| = \delta$. We are concerned about the convergence of the NSIWT method when the initial datum b is perturbed. Hereafter we will use the notation $x^{k,\delta}_{\alpha_k,r_k}$ for the solution of NSIWT (5.34) with initial datum b^δ.

The following result can be proved similarly to Theorem 1.7 in [25].

Theorem 5.30. *Under the assumptions of Corollary 5.25, if $\{\delta_k\}$ is a sequence convergent to 0 with $\delta_k \geq 0$ and such that*

$$\lim_{k\to\infty} \delta_k \cdot \sum_{j=1}^{k} \alpha_j^{-1} = 0, \tag{5.46}$$

then, $\lim_{k\to\infty} \|x^\dagger - x^{k,\delta_k}_{\alpha_k,r_k}\| = 0$.

Proof. From the definition of the method (5.33), for every given i, n, we find that

$$\begin{aligned} x^{i,\delta_k}_{\alpha_i,r_i} &= \left[(A^*A)^{\frac{r_i+1}{2}} + \alpha_i I\right]^{-1} \left((A^*A)^{\frac{r_i-1}{2}} A^* b^{\delta_k} + \alpha_i x^{i-1,\delta_k}_{\alpha_{i-1},r_{i-1}}\right) \\ &= \left\{I - \alpha_i \left[(A^*A)^{\frac{r_i+1}{2}} + \alpha_i I\right]^{-1}\right\} x^\dagger + \alpha_i \left[(A^*A)^{\frac{r_i+1}{2}} + \alpha_i I\right]^{-1} x^{i-1,\delta_k}_{\alpha_{i-1},r_{i-1}} \\ &\quad + \left[(A^*A)^{\frac{r_i+1}{2}} + \alpha_i I\right]^{-1} (A^*A)^{\frac{r_i-1}{2}} A^* \left(b^{\delta_k} - b\right), \end{aligned}$$

namely,

$$\begin{aligned} x^\dagger - x^{i,\delta_k}_{\alpha_i,r_i} &= \alpha_i \left[(A^*A)^{\frac{r_i+1}{2}} + \alpha_i I\right]^{-1} \left(x^\dagger - x^{i-1,\delta_k}_{\alpha_{i-1},r_{i-1}}\right) \\ &\quad - \left[(A^*A)^{\frac{r_i+1}{2}} + \alpha_i I\right]^{-1} (A^*A)^{\frac{r_i-1}{2}} A^* \left(b^{\delta_k} - b\right). \end{aligned}$$

Hence, by induction, for every fixed k we have

$$\begin{aligned} x^\dagger - x^{k,\delta_k}_{\alpha_k,r_k} &= \prod_{j=1}^{k} \alpha_j \left[(A^*A)^{\frac{r_j+1}{2}} + \alpha_j I\right]^{-1} x^\dagger \\ &\quad - \sum_{j=1}^{k} \alpha_j^{-1} \prod_{i=j}^{k} \alpha_i \left[(A^*A)^{\frac{r_i+1}{2}} + \alpha_i I\right]^{-1} (A^*A)^{\frac{r_j-1}{2}} A^* \left(b^{\delta_k} - b\right). \end{aligned}$$

If we set $g_{j,k}(A^*A) = \prod_{i=j}^{k} \alpha_i \left[(A^*A)^{\frac{r_i+1}{2}} + \alpha_i I\right]^{-1} (A^*A)^{\frac{r_j-1}{2}}$, then we have

$$\begin{aligned}\|g_{j,k}(A^*A)\,A^*b\|^2 &= \langle g_{j,k}(A^*A)\,A^*b, g_{j,k}(A^*A)\,A^*b\rangle \\ &= \langle g_{j,k}(AA^*)\,AA^*b, g_{j,k}(AA^*)\,b\rangle \\ &= \langle g_{j,k}(AA^*)\,(AA^*)^{1/2}\,b, g_{j,k}(A^*A)\,(AA^*)^{1/2}\,b\rangle \\ &= \|g_{j,k}(AA^*)\,(AA^*)^{1/2}\,b\|^2,\end{aligned}$$

where we used the fact that $g_{j,k}(A^*A)\,A^* = A^* g_{j,k}(AA^*)$ and that for every bounded Borel function f and h, the product $f(A)\,h(B)$ commutes if the self-adjoint operators A and B commute [116, see 12.24]. Therefore,

$$\begin{aligned}\left\|\prod_{i=j}^{k} \alpha_i \left[(A^*A)^{\frac{r_i+1}{2}} + \alpha_i I\right]^{-1} (A^*A)^{\frac{r_j-1}{2}} A^*\right\| &= \left\|\prod_{i=j}^{k} \alpha_i \left[(AA^*)^{\frac{r_i+1}{2}} + \alpha_i I\right]^{-1} (AA^*)^{\frac{r_k}{2}}\right\| \\ &= \max_{\sigma\in[0,1]} \left|\sigma^{r_j} \prod_{i=j}^{k} \frac{\alpha_i}{\sigma^{r_i+1}+\alpha_i}\right| \le 1.\end{aligned}$$

It follows that

$$\begin{aligned}\|x^\dagger - x^{k,\delta_k}_{\alpha_k,r_k}\| &\le \|\prod_{j=1}^{k} \alpha_j \left[(A^*A)^{\frac{r_j+1}{2}} + \alpha_j I\right]^{-1} x^\dagger\| + \sum_{j=1}^{k} \alpha_j^{-1}\|b^{\delta_k} - b\| \\ &= \|x^\dagger - x^k_{\alpha_k,r_k}\| + \delta_k \sum_{j=1}^{k} \alpha_k j^{-1},\end{aligned}$$

and by Corollary 5.27 and (5.46), $\|x^\dagger - x^{k,\delta_k}_{\alpha_k,r_k}\| \to 0$ for $n \to \infty$. □

5.4.2 Nonstationary iterated fractional Tikhonov

Definition 5.31 (Nonstationary iterated fractional Tikhonov)**.** *Let $\{\alpha_k\}_{k\in\mathbb{N}}$ and $\{\gamma_k\}_{k\in\mathbb{N}}$ be sequences of real numbers such that $\alpha_k > 0$ and $\gamma_k \ge 1/2$ for every k. We define the* nonstationary iterated fractional Tikhonov method (NSIFT) *as*

$$\begin{cases} x^0_{\alpha_0,\gamma_0} := 0; \\ (A^*A + \alpha_k I)^{\gamma_k} x^k_{\alpha_k,\gamma_k} := (A^*A)^{\gamma_k - 1} A^*b + [(A^*A + \alpha_k I)^{\gamma_k} - (A^*A)^{\gamma_k}]\, x^{k-1}_{\alpha_{k-1},\gamma_{k-1}}. \end{cases} \tag{5.47}$$

We denote by $x^{k,\delta}_{\alpha_k,\gamma_k}$ the k-th iteration of NSIFT if $b = b^\delta$.

Convergence analysis

Theorem 5.32. *For every $x^\dagger \in \mathcal{X}$, the* NSIFT *method* (5.47) *converges to $x^\dagger \in \mathcal{X}$ as $k \to \infty$ if and only if $\sum_k \left(\frac{\sigma^2}{\sigma^2+\alpha_k}\right)^{\gamma_k}$ diverges for every $\sigma > 0$.*

Proof. The proof follows the same steps as in Theorem 5.24. Therefore we will omit details. What follows is that

$$x^\dagger - x^k_{\alpha_k,\gamma_k} = \prod_{j=1}^{k} (A^*A + \alpha_j I)^{-\gamma_j} \left[(A^*A + \alpha_j I)^{\gamma_j} - (A^*A)^{\gamma_j} \right] x^\dagger,$$

and hence

$$\|x^\dagger - x^k_{\alpha_k,\gamma_k}\|^2 = \int_{\lambda(A^*A)} \left| \prod_{j=1}^{k} \frac{\left(\sigma^2 + \alpha_j\right)^{\gamma_j} - \sigma^{2\gamma_j}}{\left(\sigma^2 + \alpha_j\right)^{\gamma_j}} \right|^2 \mathrm{d}\langle E_{\sigma^2} x^\dagger, x^\dagger \rangle.$$

Then, the method converges if and only if

$$\lim_{k\to\infty} \prod_{j=1}^{k} \left[1 - \left(\frac{\sigma^2}{\sigma^2 + \alpha_j} \right)^{\gamma_j} \right] = 0$$

for every $\sigma > 0$. The thesis follows by Lemma 5.21. □

Corollary 5.33.

(1) Let $\lim_{j\to\infty} \gamma_j = \gamma \in [1/2, \infty)$. Then the NSIFT method converges if and only if

$$\sum_{j=1}^{k} \alpha_j^{-\gamma} = \infty.$$

More in general, if $\sup_{j\in\mathbb{N}}\{\gamma_j\} = s \in [1/2, \infty)$ and $\sum_{j=1}^{\infty} \alpha_j^{-s} = \infty$, then the NSIFT method converges.

(2) Let $\lim_{j\to\infty} \gamma_j = \infty$. If $\lim_{j\to\infty} \alpha_j = 0$ and $\lim_{j\to\infty} \alpha_j\gamma_j = l \in [0, \infty)$, then the NSIFT method converges.

Proof. (1) It is immediate noticing that

$$\sum_{j=1}^{k} \left(\frac{\sigma^2}{\sigma^2 + \alpha_j} \right)^{\gamma_j} \sim c \sum_{j=1}^{k} \left(\frac{\sigma^2}{\sigma^2 + \alpha_j} \right)^{\gamma}$$
$$\sum_{j=1}^{k} \left(\frac{\sigma^2}{\sigma^2 + \alpha_j} \right)^{\gamma_j} \geq \sum_{j=1}^{k} \left(\frac{\sigma^2}{\sigma^2 + \alpha_j} \right)^{s}.$$

(2) We observe that

$$\left(\frac{\sigma^2}{\sigma^2 + \alpha_j} \right)^{\gamma_j} = \left(1 - \frac{\alpha_j}{\sigma^2 + \alpha_j} \right)^{\gamma_j} \sim \mathrm{e}^{-\frac{\alpha_j\gamma_j}{\sigma^2+\alpha_j}} \to \mathrm{e}^{-l/\sigma^2} \neq 0$$

for $j \to \infty$. Then $\sum_{j=1}^{k} \left(\frac{\sigma^2}{\sigma^2+\alpha_j} \right)^{\gamma_j}$ diverges for every $\sigma > 0$ and the NSIFT method converges. □

Theorem 5.34. *Let $\{x^k_{\alpha_k,\gamma_k}\}_{n\in\mathbb{N}}$ be a convergent sequence of the NSIFT method, with $x^\dagger \in \mathcal{X}_\nu$ for some $\nu > 0$, and let $\{\vartheta_k\}_{k\in\mathbb{N}}$ be a divergent sequence of positive real numbers. If*

$$\lim_{k\to\infty} \vartheta_k \sigma^\nu \prod_{j=1}^{k} \left(1 - \frac{\sigma^{2\gamma_j}}{(\sigma^2+\alpha_j)^{\gamma_j}}\right) = 0 \qquad \text{for every } \sigma \in \sigma(A)\setminus\{0\};$$

$$\sup_{\sigma\in\sigma(A)\setminus\{0\}} \vartheta_k \sigma^\nu \prod_{j=1}^{k} \left(1 - \frac{\sigma^{2\gamma_j}}{(\sigma^2+\alpha_j)^{\gamma_j}}\right) \le c < \infty \qquad \text{uniformly with respect to } k,$$

then

$$\|x^\dagger - x^k_{\alpha_k,\gamma_k}\| = o\left(\vartheta_k^{-1}\right).$$

Proof. As seen in Theorem 5.26, the thesis follows easily from the Dominated Convergence Theorem. □

Corollary 5.35. *Let $\{\gamma_j\}_{j\in\mathbb{N}}$ be a sequence of positive real numbers, $\gamma_j \ge 1/2$, and let $x^\dagger \in X_\nu$ for some $\nu > 0$. If*

(i.1) $\sup_{j\in\mathbb{N}}\{\gamma_j\} = s \in [1/2,\infty)$,

(i.2) $\lim_{k\to\infty}\beta_k = \infty$,

then

$$\|x^\dagger - x^k_{\alpha_k,\gamma_n}\| = o(\beta_k^{-\frac{\nu}{2s}}) \qquad \text{if } \exists \lim_{j\to\infty}\alpha_j = \alpha \in (0,\infty], \tag{5.49}$$

$$\|x^\dagger - x^k_{\alpha_k,\gamma_k}\| = o(\tilde\beta_k^{-\frac{\nu}{2s}}) \qquad \text{otherwise}, \tag{5.50}$$

where we defined

$$\beta_k = \sum_{j=1}^{k} \alpha_j^{-s}, \qquad \tilde\beta_k = \sum_{j=1}^{k} \frac{1}{1+\alpha_j^s}.$$

Proof. See Corollary 5.27. □

Analysis of convergence for perturbed data

Theorem 5.36. *Under the assumptions of Corollary 5.33, if $\{\delta_k\}$ is a sequence convergent to 0 with $\delta_k \ge 0$ and such that*

$$\lim_{k\to\infty} \delta_k \cdot \sum_{j=1}^{k} \alpha_j^{-\gamma_j} = 0,$$

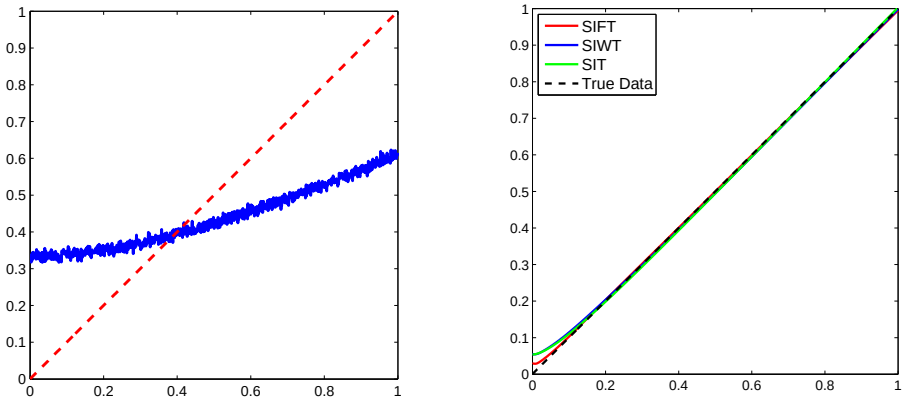
then, $\lim_{k\to\infty}\|x^\dagger - x^{k,\delta_k}_{\alpha_k,\gamma_k}\| = 0$.

Proof. Here is a sketch of the proof, since it follows step by step from the proof of Theorem 5.30. If we set

$$\psi_j(A^*A) := [(A^*A+\alpha_j I)^{\gamma_j} - (A^*A)^{\gamma_j}]$$
$$\phi_j(A^*A) := \psi_j(A^*A)\,[A^*A+\alpha_j I]^{-\gamma_j},$$

then from (5.47) it is possible to show that

$$x^\dagger - x^{k,\delta_k}_{\alpha_k,\gamma_k} = \prod_{j=1}^{k}\phi_j(A^*A)\,x^\dagger - \sum_{j=1}^{k}\psi_j(A^*A)^{-1}\prod_{i=j}^{k}\phi_i(A^*A)\,(A^*A)^{\gamma_j-1}A^*\left(b^{\delta_k}-b\right),$$



(a) (b)

FIGURE 5.1: `Foxgood` test problem: (a) the true solution (dashed curve) and the observed data (solid curve), (b) approximated solutions by SIFT with $\gamma = 0.8$ and $\alpha = 10^{-3}$, SIWT with $r = 0.6$ and $\alpha = 10^{-2}$, and SIWT with $r = 1$ and $\alpha = 10^{-3}$.

for every integer n and for every perturbed data $b^{\delta_k} = b + \delta_k \eta$. Owing to the equality

$$\left\| \prod_{i=j}^{k} \phi_i \left(A^*A\right) \left(A^*A\right)^{\gamma_j - 1} A^* \right\| = \left\| \prod_{i=j}^{k} \phi_i \left(AA^*\right) \left(AA^*\right)^{\gamma_j - 1} \left(AA^*\right)^{1/2} \right\|,$$

we deduce

$$\begin{aligned} \|x^\dagger - x^{k,\delta_k}_{\alpha_k,\gamma_k}\| &\leq \|x^\dagger - x^k_{\alpha_k,\gamma_k}\| + \delta_k \sum_{j=1}^{k} \left\| \psi_j \left(A^*A\right)^{-1} \right\| \\ &= \|x^\dagger - x^k_{\alpha_k,\gamma_k}\| + \delta_k \sum_{j=1}^{k} \alpha_j^{-\gamma_j}. \end{aligned}$$

□

5.5 Numerical examples

We now give few selected examples with a special focus on the nonstationary iterations proposed in this chapter. For a larger comparison between fractional and classical Tikhonov refer to [69, 86, 95]. To produce our results we used MATLAB 8.1.0.604 using a laptop pc with processor Intel iCore i5-3337U with 6 GB of RAM running Windows 8.1.

In all the examples we add to the noise-free right-hand side vector b white Gaussian noise with noise level ξ.

As a stopping criterion for the methods we used the discrepancy principle (2.19) with $\tau = 1.01$. This criterion stops the iterations when the norm of the residual reaches the norm of the noise so that the latter is not reconstructed.

To compare the restorations with the different methods, we consider both the visual representation and the relative restoration error for the computed approximation $\hat{x}$.

α	Method	r/γ				
		0.4	0.6	0.8	1	1.2
5×10^{-2}	SIFT	337.09(7)	0.02498(13)	0.03481(19)	0.03752(29)	0.03838(43)
	SIWT	0.02589(9)	0.03202(13)	0.03609(19)	0.03752(29)	0.03932(43)
10^{-2}	SIFT	320.85(3)	0.02048(5)	0.02633(7)	0.03731(7)	0.03783(9)
	SIWT	0.01697(3)	0.01818(5)	0.03361(5)	0.03731(7)	0.03672(11)
5×10^{-3}	SIFT	423.37(3)	0.02216(3)	0.02190(5)	0.03102(5)	0.03723(5)
	SIWT	0.02421(3)	0.01573(3)	0.03186(3)	0.03103(5)	0.03347(7)
10^{-3}	SIFT	402.97(1)	0.02299(1)	**0.00698(3)**	0.01756(3)	0.02443(3)
	SIWT	0.06403(1)	0.02210(1)	0.02528(1)	0.01756(3)	0.02736(3)
5×10^{-4}	SIFT	531.72(1)	0.02119(1)	0.01729(1)	0.02507(1)	0.03119(1)
	SIWT	0.10518(1)	0.04506(1)	0.01482(1)	0.02507(1)	0.02086(3)
10^{-4}	SIFT	1012.2(1)	0.07246(1)	0.04229(1)	0.02704(1)	0.01675(1)
	SIWT	0.25927(1)	0.13000(1)	0.07213(1)	0.02704(1)	0.01154(1)

TABLE 5.1: `Foxgood` test problem: RRE for SIWT and SIFT for different choices of α, r, and γ. The smallest error is shown in boldface.

Foxgood This test case is the so-called `Foxgood` in the toolbox REGULARIZATION TOOL [83] using 1024 points. We have added a noise vector with $\xi = 0.02$ to the observed signal. In Figure 5.1(a) the true signal and the measured data can be seen.

In Table 5.1 we show the relative errors with different choices of α, r and γ. In brackets we report the iteration at which the discrepancy principle stopped the method. Note that SIFT with $\gamma = 1$ and SIWT with $r = 1$ are exactly the classical Tikhonov method and hence produce the same result. Figure 5.1(b) shows the reconstruction for SIFT with $\gamma = 0.8$ and $\alpha = 10^{-3}$, SIWT with $r = 0.6$ and $\alpha = 10^{-2}$, and SIWT with $r = 1$ (classical Iterated Tikhonov) with $\alpha = 10^{-3}$.

From these results, using both fractional and weighted iterated Tikhonov, we can see that we can obtain better restorations than with the classical version. However, in order to obtain such results, one has to evaluate α very carefully. Indeed α does not only affects the convergence speed, but also the quality of the restoration: a small perturbation in α can lead to quite different restoration errors. The nonstationary version of the methods can help also to avoid such a careful and often difficult estimation.

For the nonstationary iterations we assume the regularization parameter α_k at each iteration be given according to the geometric sequence

$$\alpha_k = \alpha_0 q^k, \qquad q \in (0,1), \qquad k = 1, 2, \dots. \tag{5.51}$$

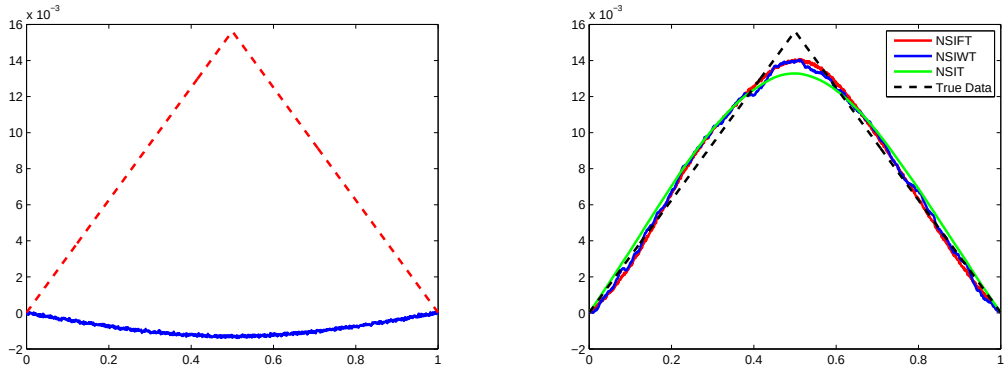
Setting $r_k = 0.6$ and $\gamma_k = 0.8$, Table 5.2 shows that NSIFT and NSIWT provide a relative error lower than the classical nonstationary iterated Tikhonov (IT_{NS}). Finally, since NSIFT and NSIWT allow a nonstationary choice also for r_k and γ_k, in Table 5.2 we report the results for the following nonincreasing sequences

$$r_k = \gamma_k = \begin{cases} 1 - \frac{k-1}{100} & k < 50, \\ \frac{1}{2} & \text{otherwise.} \end{cases} \tag{5.52}$$

Again both NSIWT and NSIFT are able to get better results than IT_{NS}. Even tough the errors are not as good as those for the best choices $r_k = 0.6$ and $\gamma_k = 0.8$, the choice (5.52) stresses the robustness of our nonstationary iterations.

α_0	Method	q		
		0.7	0.8	0.9
10^{-1}	NSIFT ($\gamma_k = 0.8$)	0.024453(9)	0.030868(11)	0.028849(17)
	NSIWT ($r_k = 0.6$)	0.025223(7)	0.027628(9)	0.028534(13)
	IT$_{NS}$	0.035162(9)	0.031627(13)	0.036472(19)
	NSIFT (γ_k in (5.52))	0.032489(9)	0.027974(13)	0.037199(17)
	NSIWT (r_k in (5.52))	0.031493(9)	0.027436(13)	0.036059(17)
10^{-2}	NSIFT ($\gamma_k = 0.8$)	**0.014781(5)**	0.021687(5)	0.028709(5)
	NSIWT ($r_k = 0.6$)	**0.014503(3)**	0.021501(3)	0.028396(3)
	IT$_{NS}$	0.024838(5)	0.030866(5)	0.028835(7)
	NSIFT (γ_k in (5.52))	0.023848(5)	0.030002(5)	0.027636(7)
	NSIWT (r_k in (5.52))	0.023482(5)	0.029638(5)	0.027366(7)

TABLE 5.2: `Foxgood` test problem: RRE for NSIWT and NSIFT with the nonstationary α_k in (5.51) and different choices of r_k and γ_k (IT$_{NS}$ is $r_k = \gamma_k = 1$). The smallest error is shown in boldface.



(a) (b)

FIGURE 5.2: `Deriv2` test problem: (a) the true solution (dashed curve) and the observed data (solid curve), (b) approximated solutions.

Deriv2 We consider the test problem `deriv2(·,3)` in the toolbox REGULARIZATION TOOL [83] using 1024 points. For the noise vector it holds $\xi = 0.05$. In Figure 5.2(a) we can see the measured data and the true signal. We compare NSIWT and NSIFT with the IT$_{NS}$.

Firstly, α_k is defined by the classical choice in (5.51). Table 5.3 shows the results for different choices of r_k and γ_k. Note that NSIWT and NSIFT usually outperform IT$_{NS}$. Nevertheless, our nonstationary iterations allow also unbounded sequences of r_k and γ_k. Therefore, according to Proposition 5.29, we set

$$\alpha_k = \frac{1}{k!}, \qquad r_k = \frac{k}{10}, \qquad \gamma_k = \frac{k}{2}. \tag{5.53}$$

Table 5.4 shows that the relative restoration error obtained with the unbounded sequences r_k and γ_k in (5.53) is lower than the best one (according to Table 5.3), obtained by IT$_{NS}$ by employing the geometric sequence (5.51) for α_k. The computed approximations are also compared in Figure 5.2(b), where we note a better restoration of the corner for NSIWT and NSIFT.

Blur We consider the test problem `blur(·,·,·)` in the toolbox REGULARIZATION TOOL by P. Hansen [83]. This is a two dimensional deblurring problem, the true solution is a 40×40

α_0	Method	q		
		0.7	0.8	0.9
10^{-1}	NSIFT ($\gamma_k = 0.8$)	0.08981(11)	0.09394(13)	0.09445(19)
	NSIWT ($r_k = 0.6$)	0.08051(13)	0.09181(17)	0.09401(29)
	IT$_{NS}$	0.08502(15)	0.09175(21)	0.09466(37)
	NSIFT (γ_k in (5.52))	0.09428(13)	0.09089(19)	0.09327(29)
	NSIWT (r_k in (5.52))	0.09073(13)	0.08648(19)	0.09199(29)
10^{-2}	NSIFT ($\gamma_k = 0.8$)	0.09114(5)	0.08953(7)	0.08998(9)
	NSIWT ($r_k = 0.6$)	**0.07807(7)**	0.09411(7)	0.09183(11)
	IT$_{NS}$	0.08183(9)	0.09174(11)	0.09379(17)
	NSIFT (γ_k in (5.52))	**0.07839(9)**	0.08721(11)	0.09246(15)
	NSIWT (r_k in (5.52))	0.09399(7)	0.08389(11)	0.08990(15)

TABLE 5.3: `Deriv2` test problem: RRE for NSIWT and NSIFT with the non-stationary α_k in (5.51) and different choices of r_k and γ_k (IT$_{NS}$ is $r_k = \gamma_k = 1$). The smallest error is shown in boldface.

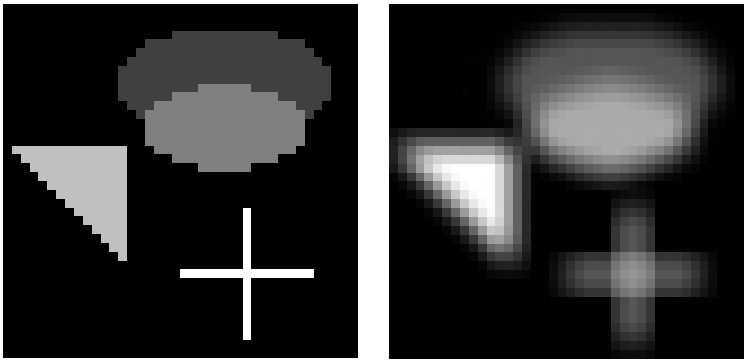
	NSIFT	NSIWT	IT$_{NS}$
Error	0.054831(9)	0.059211(7)	0.081835(9)

TABLE 5.4: `Deriv2` test problem: relative restoration errors for NSIFT and NSIWT with parameters in (5.53) and IT$_{NS}$ with $\alpha_k = 0.01 \cdot 0.7^k$.

image, the blurring operator is a symmetric BTTB with bandwidth 6. This blur is created by a truncated Gaussian point spread function with variance 2. For the noise vector it holds $\nu = 0.005$. Figure 5.3(a) shows the true image while the observed image is in Figure 5.3(b).

Firstly, α_k is defined by the classical choice in (5.51). Table 5.5 provides the results for a good stationary choice of r_k and γ_k. Note that NSIWT and NSIFT usually outperform IT$_{NS}$. Table 5.6 shows that the relative restoration error obtained with the unbounded sequences r_k and γ_k in (5.53) is lower than the best one (according to Table 5.5), obtained by the stationary choice of r_k and γ_k. We note that NSIWT and NSIFT are less sensitive than IT$_{NS}$ to an appropriate choice of α_0 and q. In particular using r_k and γ_k in (5.53), NSIWT and NSIFT do not need any parameter estimation and the computed solutions have a relative restoration error lower than IT$_{NS}$ with the best parameter setting (see Table 5.5) and they provide also a better reconstruction, in particular of the edges, see Figure 5.4.

Finally, note that for the IT$_{NS}$ a nondecreasing sequence of α_k could be considered instead of the geometric sequence (5.51), see [46]. Nevertheless, this strategy requires a proper choice



(a) (b)

FIGURE 5.3: `Blur` test problem: (a) the true image, (b) the measured data.

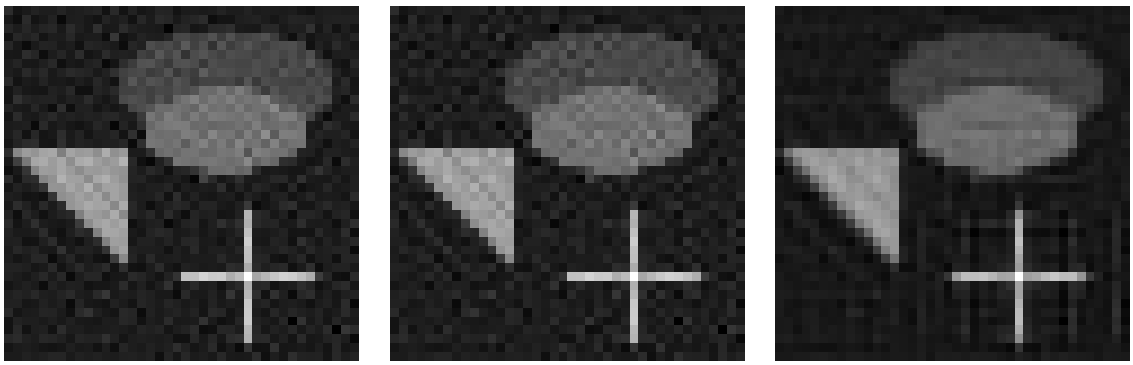
α_0	Method	q		
		0.7	0.8	0.9
10^{-1}	NSIFT ($\gamma_k = 0.5$)	0.19970(9)	0.19526(13)	0.19847(17)
	NSIWT ($r_k = 0.2$)	0.18936(7)	**0.18920(9)**	0.19732(11)
	IT_{NS}	0.19816(15)	0.21786(20)	0.28703(20)
10^{-2}	NSIFT ($\gamma_k = 0.5$)	0.19398(5)	0.19962(5)	0.19595(7)
	NSIWT ($r_k = 0.2$)	0.20822(3)	0.19547(3)	0.19109(3)
	IT_{NS}	0.19518(9)	0.20531(11)	0.20747(17)

TABLE 5.5: `Blur` test problem: RRE for NSIWT and NSIFT with the nonstationary α_k in (5.51). The smallest error is shown in boldface.

	NSIFT	NSIWT	IT_{NS}
Error	0.19335(10)	**0.18765(8)**	0.19518(9)

TABLE 5.6: `Blur` test problem: relative restorations errors for NSIFT and NSIWT with parameters in (5.53) and IT_{NS} with $\alpha_k = 0.01 \cdot 0.7^k$. The smallest error is shown in boldface.

of α_0 and this is out of the scope of this paper, but it could be investigated in the future in connection with our fractional and weighted variants. A further development of our iterative schemes is in the direction of the nonstationary preconditioning strategy in [49], which is inspired by an approximated solution of the IT_{NS} and hence could be investigated also in a fractional framework.



(a) (b) (c)

FIGURE 5.4: `Blur` test problem reconstructions: (a) NSIFT and (b) NSIWT with parameters in (5.53), (c) IT_{NS} with $\alpha_k = 0.01 \cdot 0.7^k$.

Chapter 6

Approximated Iterated Tikhonov: some extensions

As in the previous chapter we treat the problem in the continuous setting, i.e., when $A : \mathcal{X} \to \mathcal{Y}$ is a linear operator between the Hilbert spaces $\mathcal{X}$ and $\mathcal{Y}$. In [49] the authors developed an iterative method with a nonstationary preconditioner, that can be seen as an approximated iterated Tikhonov regularization. In particular they considered an operator C which is spectrally equivalent to A (see Assumption 6.1 in the next section) and form the preconditioner at step k as

$$C^*(CC^* + \alpha_k I)^{-1} \approx A^*(AA^* + \alpha_k I)^{-1},$$

where α_k is determined by a damped version of the discrepancy principle. In this way they are able to both achieve fast computation, by wisely choosing the structure of C, and have a parameter free method. The estimation of the parameter α_k can be difficult. For example, in the mentioned geometric sequence there are two parameter to be estimated: α_0 and q. Even though small changes in either α_0 and q have only a limited effect on the quality of the reconstruction, an imprudent choice can still lead to poor results. Because, roughly speaking, they are approximating the operator A with C we will refer to this method as *Approximated Iterated Tikhonov (AIT)*. Another extension of the AIT method has been proposed in [44]. In the this work the authors consider the case of image deblurring and use the eigenvalue of the preconditioner generated by AIT as a generating function for a structured preconditioner inside an iterative refinement technique.

In this chapter we want to add some features to the algorithm in [49] and test the resulting methods on image deblurring. Since we have shown in Chapter 4 that the introduction of a regularization operator L in the IT iterations can improve the quality of the obtained reconstructions, at first we study the introduction of L in place of I in AIT. We call the resulting method *AIT-GP (Approximated Iterated Tikhonov with General Penalty term)*. If we know that $x^\dagger$ lies in some closed and convex $\Omega \subset \mathcal{X}$, we constrain the algorithm in order to get $x_k \in \Omega,\ \forall k$. Hence, we modify AIT introducing the metric projection into Ω. We refer to this method as *APIT (Approximated Projected Iterated Tikhonov)*. For both the previous generalizations of the iterative method proposed in [49], namely AIT-GP and APIT, we prove that the new iterations are convergent in the noise-free case and that are regularization methods in the noisy case. Finally, we combine the regularization term and the projection into Ω developing a third algorithm called *APIT-GP (Approximated Projected Iterated Tikhonov with General Penalty term)*.

This chapter is structured as follows. Section 6.1 describes the AIT method proposed in [49]. In Sections 6.2 and 6.3 we define and study the theoretical properties of our new three iterative regularization methods. Finally, in Section 6.4 the proposed methods are applied to the image deblurring problem and compared with other methods proposed in the literature.

6.1 Approximated Iterated Tikhonov

We now describe the preconditioned iteration proposed in [49]. We need the following assumption that it will be necessary also for our algorithms in the next sections.

Assumption 6.1. *Let C be a linear operator such that*

$$\|(C-A)z\| \leq \rho \|Az\|, \qquad \forall z \in \mathcal{X}, \tag{6.1}$$

for some $0 < \rho < \frac{1}{2}$. We say that C is spectrally equivalent to A.

Under this assumption it holds a preliminary result useful for the convergence analysis.

Define the residual at the k-th step as

$$r_k = b^\delta - Ax_k,$$

then the following holds

Lemma 6.1 ([49]). *Assume that* (2.2) *and Assumption 6.1 hold. If $\tau_k = \|r_k\|/\delta > \tau_* = (1+\rho)/(1-2\rho)$, then it follows that*

$$\|r_k - Ce_k\| \leq \left(\rho + \frac{1+\rho}{\tau_k}\right)\|r_k\| < (1-\rho)\|r_k\|.$$

Algorithm 6.1 (Approximated Iterated Tikhonov (AIT)). *Let $x_0 \in \mathcal{X}$ be fixed. Choose $\tau = \frac{1+2\rho}{1-2\rho}$ with ρ as in* (6.1)*, and fix $q \in [2\rho, 1]$.*

$k = 0,\ r_0 = b^\delta - Ax_0, \tau_0 = \dfrac{\|r_k\|}{\delta}$

`While` $\|r_k\| > \tau\delta$

$\tau_k = \dfrac{\|r_k\|}{\delta}$

$q_k = \max\left\{q, 2\rho + \dfrac{1+\rho}{\tau_k}\right\}$

`compute` α_k `such that` $\left\|r_k - CC^*(CC^* + \alpha_k I)^{-1} r_k\right\| = q_k \|r_k\|$

$h_k = C^*(CC^* + \alpha_k I)^{-1} r_k$

$x_{k+1} = x_k + h_k$

$r_{k+1} = b^\delta - Ax_{k+1}$

We summarize the main theoretical results proved in [49] about the convergence and the monotonic decrease of the norm of the error for AIT. We denote the iteration error e_k by

$$e_k = x^\dagger - x_k.$$

Proposition 6.2 ([49]). *Under Assumption 6.1, while $\|r_k\| > \tau\delta$, with $\tau = (1+2\rho)/(1-2\rho)$ the norm of the reconstruction error e_k decreases monotonically, namely $\|e_{k+1}\| \leq \|e_k\|$, for $k = 0, 1, \ldots$.*

Corollary 6.3 ([49]). *With the notation and assumptions of Proposition 6.2, it holds*

$$\|e_0\|^2 \geq 2\rho \sum_{k=0}^{k^\delta - 1} \left\|(CC^* + \alpha_k I)^{-1} r_k\right\| \|r_k\| \geq c \sum_{k=0}^{k^\delta - 1} \|r_k\|^2.$$

Theorem 6.4 ([49]). *Assume that the data are exact, i.e., $\delta = 0$, and that x_0 is not a solution of problem* (2.3)*. Then the sequence $(x_k)_k$ converges as $k \to \infty$ to the solution of* (2.3) *which is nearest to x_0.*

Theorem 6.5 ([49]). *Let $\delta \mapsto b^\delta$ be a function from $\mathbb{R}^+$ to $\mathcal{Y}$ such that* (2.2) *holds true for all $\delta > 0$. Under Assumption 6.1, for two fixed parameters τ and q, denote by x^δ the resulting approximation obtained with AIT. Then for $\delta \to 0$ we have that $x^\delta \to x_0^\dagger$ which is the nearest solution of* (2.3) *to x_0.*

In [49] the choice of x_0 was not deeply investigated. For many iterative regularization methods, like Krylov methods, the null vector is usually a good choice for x_0. Nevertheless, for AIT this is not a good choice and setting $x_0 = A^* b^\delta$, which is the initial solution subspace vector for LSQR, usually provides better results. This is confirmed by several numerical experiments with image deblurring problems and follows from the next observation. The approximation of A by C is motivated by the fact that the error equation, used for the iterative refinement, allows a slight misfit due to the noise already present in the problem. If we choose $x_0 = 0$ then $r_0 = b^\delta$ and $x_1 = x_0 + C^*(CC^* + \alpha_k I)^{-1} b^\delta$ which is exactly the Tikhonov solution for the operator C instead of A. Although, from a theoretical point of view, this should not be a problem, numerically this can lead to some issues. For example if C does not approximate well A then x_1 could contain large error components. These components may be hard to reduce in the following iterations to the point of producing slightly worse reconstructions than the ones obtained by $x_0 = A^* b^\delta$.

6.2 Approximated Iterated Tikhonov with general penalty term (AIT-GP)

In this section we combine the idea of AIT with the generalized iterated Tikhonov method (Algortihm 4.1), i.e., we introduce the regularization operator L in Algorithm 6.1. We need a couple of assumptions that link the matrix L with A and C similarly to the basic assumption (2.9).

Assumption 6.2. *Let L and C be two linear operators such that*

(i) $C|_{\mathcal{N}(L)} = A|_{\mathcal{N}(L)}$;

(ii) L and C are diagonalized by the same unitary transformation.

Assumption 6.2(ii) is restrictive, but it is needed for the proofs that follows. This kind of requirements can be satisfied for certain choices of C and L and in particular for certain classes of structured matrices. In Section 6.4 we show an example.

Note that thanks to (6.1) $\mathcal{N}(A) = \mathcal{N}(C)$ and hence (2.9) implies that $\mathcal{N}(L) \cap \mathcal{N}(C) = \{0\}$.

Remark 6.6. *Under the assumption* (ii) *on C and L we have that*

$$C : \mathcal{X} \to \mathcal{X},$$
$$L : \mathcal{X} \to \mathcal{X}.$$

It is indeed possible to choose an $L : \mathcal{X} \to \mathcal{Z}$ and then transform it into an operator to $\mathcal{X}$ either via an appropriate zero padding *or using its QR factorization, see Section 4.2.2. However, it can be challenging, if not impossible, to prove that Assumption 6.2*(ii) *holds after the transformation.*

We define the orthogonal projection over $\mathcal{N}(L)$

$$P_{\mathcal{N}(L)} = I - L^\dagger L$$

and the orthogonal projection over $\mathcal{N}(L)^\perp$

$$P_{\mathcal{N}(L)^\perp} = L^\dagger L.$$

From Remark 6.6 and Assumption 6.2(ii) we have the following

Lemma 6.7. *Let L and C be operator that satisfy Assumption 6.2(ii) and let $L_C^\dagger$ be the operator defined in* (2.17)*, then it holds*

(i) $C^\dagger C$ *commutes with* $L^\dagger L$*;*

(ii) $(I - L^\dagger L)C = C(I - L^\dagger L)$*;*

(iii) $(C(I - L^\dagger L))^\dagger = (I - L^\dagger L)C^\dagger$.

Proof. By Assumption 6.2(ii) there exists a unitary transformation F such that

$$\begin{aligned} C &= F\Gamma F^*, \\ L &= F\Lambda F^*. \end{aligned}$$

From [108, Lemma 1·6], see also [16, Theorem 2.2.2], it holds that

$$\begin{aligned} C^\dagger &= F\Gamma^\dagger F^*, \\ L^\dagger &= F\Lambda^\dagger F^*. \end{aligned}$$

Thus,

$$\begin{aligned} C^\dagger C &= F\Gamma^\dagger \Gamma F^*, \\ L^\dagger L &= F\Lambda^\dagger \Lambda F^*. \end{aligned} \tag{6.2}$$

From (6.2) the proof of point (i) comes immediately. In fact

$$\begin{aligned} C^\dagger C L^\dagger L &= F\Gamma^\dagger \Gamma F^* F\Lambda^\dagger \Lambda F^* \\ &= F\Gamma^\dagger \Gamma \Lambda^\dagger \Lambda F^* \\ &= F\Lambda^\dagger \Lambda \Gamma^\dagger \Gamma F^* \\ &= F\Lambda^\dagger \Lambda F^* F\Gamma^\dagger \Gamma F^* \\ &= L^\dagger L C^\dagger C, \end{aligned}$$

where we have used the fact that diagonal operators commute with each other.

We move now to point (ii). From (6.2) we have that

$$(I - L^\dagger L) = (F^*F - F\Lambda^\dagger \Lambda F^*) = F^*(I - \Lambda^\dagger \Lambda)F.$$

We can then write

$$\begin{aligned} (I - L^\dagger L)C &= F(I - \Lambda^\dagger \Lambda)FF^*\Gamma F^* \\ &= F(I - \Lambda^\dagger \Lambda)\Gamma F^* \\ &= F\Gamma F^* F(I - \Lambda^\dagger \Lambda)F \\ &= C(I - L^\dagger L), \end{aligned}$$

proving point (ii).

Finally we can prove point (iii). In order to do that we show the four properties that characterize the Moore-Penrose pseudo-inverse. In particular X is the Moore-Penrose pseudo-inverse of A if and only if

$$AXA = A, \quad XAX = X, \quad (AX)^* = AX, \quad (XA)^* = XA,$$

see [108].

We now prove the above four properties with $A = C(I - L^\dagger L)$ and $X = (I - L^\dagger L)C^\dagger$.

$AXA = A$. Using the fact that $(I - L^\dagger L)(I - L^\dagger L) = (I - L^\dagger L)$, point (ii) and the properties of the pseudo-inverse $C^\dagger$ we have

$$\begin{aligned}\left[C(I - L^\dagger L)\right]\left[(I - L^\dagger L)C^\dagger\right]\left[C(I - L^\dagger L)\right] &= C(I - L^\dagger L)C^\dagger C(I - L^\dagger L)\\ &= (I - L^\dagger L)CC^\dagger C(I - L^\dagger L)\\ &= (I - L^\dagger L)C(I - L^\dagger L)\\ &= C(I - L^\dagger L)(I - L^\dagger L)\\ &= C(I - L^\dagger L).\end{aligned}$$

$XAX = X$. Considering also point (i) we have

$$\begin{aligned}\left[(I - L^\dagger L)C^\dagger\right]\left[C(I - L^\dagger L)\right]\left[(I - L^\dagger L)C^\dagger\right] &= (I - L^\dagger L)C^\dagger C(I - L^\dagger L)C^\dagger\\ &= (I - L^\dagger L)(I - L^\dagger L)C^\dagger CC^\dagger\\ &= (I - L^\dagger L)C^\dagger\end{aligned}$$

$(AX)^* = AX$. Noting that $(C^\dagger C)^* = C^\dagger C$ and $(L^\dagger L)^* = L^\dagger L$ by the properties of the pseudo-inverse we get

$$\begin{aligned}\left(\left[C(I - L^\dagger L)\right]\left[(I - L^\dagger L)C^\dagger\right]\right)^* &= \left(C(I - L^\dagger L)C^\dagger\right)^*\\ &= \left((I - L^\dagger L)CC^\dagger\right)^*\\ &= CC^\dagger(I - L^\dagger L)\\ &= (I - L^\dagger L)CC^\dagger\\ &= C(I - L^\dagger L)C^\dagger\\ &= \left[C(I - L^\dagger L)\right]\left[(I - L^\dagger L)C^\dagger\right].\end{aligned}$$

$(XA)^* = XA$. Analogously we get

$$\begin{aligned}\left(\left[(I - L^\dagger L)C^\dagger\right]\left[C(I - L^\dagger L)\right]\right)^* &= \left((I - L^\dagger L)(I - L^\dagger L)C^\dagger C\right)^*\\ &= \left((I - L^\dagger L)C^\dagger C\right)^*\\ &= C^\dagger C(I - L^\dagger L)\\ &= C^\dagger C(I - L^\dagger L)(I - L^\dagger L)\\ &= \left[(I - L^\dagger L)C^\dagger\right]\left[C(I - L^\dagger L)\right].\end{aligned}$$

Which concludes the proof of point (iii). □

We are now in the position of proving

Lemma 6.8. *With the same assumptions and notations of Lemma 6.7 it holds*

$$L_C^\dagger = L^\dagger$$

Proof. Let us write the expression for $L_C^\dagger$ and use the results shown in Lemma 6.7

$$\begin{aligned} L_C^\dagger &= (I - (C(I - L^\dagger L))^\dagger C)L^\dagger \\ &= (I - (I - L^\dagger L)C^\dagger C)L^\dagger \\ &= (I - C^\dagger C + L^\dagger L C^\dagger C)L^\dagger \\ &= L^\dagger - C^\dagger C L^\dagger + L^\dagger L C^\dagger C L^\dagger \\ &= L^\dagger - C^\dagger C L^\dagger + C^\dagger C L^\dagger L L^\dagger \\ &= L^\dagger - C^\dagger C L^\dagger + C^\dagger C L^\dagger \\ &= L^\dagger \end{aligned}$$

□

We define

$$\overline{C} = C L_C^\dagger = C L^\dagger. \tag{6.3}$$

Algorithm 6.2 (Approximated Iterated Tikhonov with General Penalty term (AIT-GP))**.** *Let L and C be linear operators that fulfill Assumptions 6.1 and 6.2 for a fixed $0 < \rho \leq \frac{1}{2}$.*

Let $x_0 \in \mathcal{X}$ be fixed. Choose $\tau = \frac{1+2\rho}{1-2\rho}$ with ρ from (6.1)*, and fix $q \in [2\rho, 1]$.*

$k = 0,\ r_0 = b^\delta - Ax_0,\ \tau_0 = \dfrac{\|r_k\|}{\delta}$

`While` $\|r_k\| > \tau\delta$

$\tau_k = \dfrac{\|r_k\|}{\delta}$

$q_k = \max\left\{q, 2\rho + \dfrac{1+\rho}{\tau_k}\right\}$

`compute` α_k `such that` $\left\|r_k - CC^*(CC^* + \alpha_k LL^*)^{-1} r_k\right\| = q_k \|r_k\|$

$h_k = C^*(CC^* + \alpha_k LL^*)^{-1} r_k$

$x_{k+1} = x_k + h_k$

$r_{k+1} = b^\delta - Ax_{k+1}$

Note that, by construction of α_k, it holds for all k that

$$\|r_k - Ch_k\| = q_k \|r_k\|, \tag{6.4}$$

We refer to Algorithm 6.2 as *Approximated Iterated Tikhonov with General Penalty term* (AIT-GP) since this method can be seen as a preconditioned iterative method whose preconditioner is obtained by approximated Tikhonov with a general regularization operator L.

We define, as in Subsection 2.2.1,

$$\begin{cases} h_k^{(0)} = (C(I - L^\dagger L))^\dagger r_k \\ \overline{r}_k = r_k - C h_k^{(0)}. \end{cases} \tag{6.5}$$

Note that if L is invertible then $\overline{r}_k = r_k$.

Lemma 6.9. *Let $\overline{r}_k$ be defined in* (6.5)*, then it holds*

$$\left\| L^\dagger r_k \right\| = \left\| L^\dagger \overline{r}_k \right\|.$$

Proof. From the definition of $\overline{r}_k$ and $h_k^{(0)}$ in (6.5) it follows that

$$\left\| L^\dagger \overline{r}_k \right\| = \left\| L^\dagger (r_k - C h_k^{(0)}) \right\| = \left\| L^\dagger r_k - L^\dagger C (C(I - L^\dagger L))^\dagger r_k \right\|$$

proving that $L^\dagger C(C(I - L^\dagger L))^\dagger = 0$ will conclude the proof. Consider the results in Lemma 6.7

$$L^\dagger C(C(I - L^\dagger L))^\dagger = L^\dagger C(I - L^\dagger L)C^\dagger = L^\dagger (I - L^\dagger L) C C^\dagger.$$

Note that, since $L : \mathcal{X} \to \mathcal{X}$ it holds that $\mathcal{N}(L^\dagger) = \mathcal{N}(L)$ thus, being $(I - L^\dagger L) = P_{\mathcal{N}(L)}$, we have that $L^\dagger (I - L^\dagger L) = 0$. Using this last equality we have that

$$L^\dagger C(C(I - L^\dagger L))^\dagger = L^\dagger (I - L^\dagger L) C C^\dagger = 0,$$

which concludes the proof.

In particular we also obtained that $h_k^{(0)} \in \mathcal{N}(L)$. □

Lemma 6.10. *Let $\overline{r}_k$ and $\overline{C}$ be defined in* (6.5) *and* (6.3)*, respectively, and define*

$$\overline{h}_k := L h_k.$$

Then it holds

$$\left\| \overline{r}_k - \overline{C h_k} \right\| = \left\| r_k - C h_k \right\|.$$

Proof. This results has been shown in [59], we give here a proof with our notation for completeness.

Form [59] we know that

$$h_k = L^\dagger \overline{C}^* (\overline{C C}^* + \alpha_k I)^{-1} \overline{r}_k + h_k^{(0)},$$

and so, since $\overline{C}^* (\overline{C C}^* + \alpha_k I)^{-1} \overline{r}_k \in \mathcal{N}(L)^\perp$ for construction of $\overline{r}_k$ and $\overline{C}$ and $h_k^{(0)} \in \mathcal{N}(L)$ (see Lemma 6.9), we get

$$\overline{h}_k = L h_k = L L^\dagger \overline{C}^* (\overline{C C}^* + \alpha_k I)^{-1} \overline{r}_k + L h_k^{(0)} = \overline{C}^* (\overline{C C}^* + \alpha_k I)^{-1} \overline{r}_k,$$

and so it holds

$$\overline{h}_k = \overline{C}^* (\overline{C C}^* + \alpha_k I)^{-1} \overline{r}_k. \tag{6.6}$$

Moreover,

$$\left\| \overline{r}_k - \overline{C h_k} \right\| = \left\| r_k - C h_k \right\|, \tag{6.7}$$

in fact

$$\left\|\overline{r}_k - \overline{Ch}_k\right\| = \left\|r_k - Ch_k^{(0)} - \overline{Ch}_k\right\| = \left\|r_k - C\left(L^\dagger \overline{h}_k + h_k^{(0)}\right)\right\| = \|r_k - Ch_k\|,$$

where, in the last step, we have used the definition of $\overline{C} = CL^\dagger$. □

Now we divide the space $\mathcal{X} = \mathcal{N}(L) \oplus \mathcal{N}(L)^\perp$ and we analyze the behavior of Algorithm 6.2 on each subspace.

We call

$$e_k^\perp = P_{\mathcal{N}(L)^\perp}(e_k) = P_{\mathcal{N}(L)^\perp}(x^\dagger) - P_{\mathcal{N}(L)^\perp}(x_k)$$

and

$$e_k^{(0)} = P_{\mathcal{N}(L)}(e_k) = P_{\mathcal{N}(L)}(x^\dagger) - P_{\mathcal{N}(L)}(x_k).$$

On the two subspaces the Algorithm 6.2 has different behaviors.

This is analysis is related to the one we performed in Chapter 4 for the GIT algorithm.

First, in Remark 6.11 we concentrate on the space $\mathcal{N}(L)$.

Remark 6.11. *Let us consider the projection onto $\mathcal{N}(L)$ of the very first iteration*

$$P_{\mathcal{N}(L)}(x_1) = P_{\mathcal{N}(L)}(x_0 + h_0) = P_{\mathcal{N}(L)}(x_0) + P_{\mathcal{N}(L)}(h_0),$$

since $h_k = L^\dagger \overline{h}_k + h_k^{(0)}$, we get

$$P_{\mathcal{N}(L)}(h_0) = P_{\mathcal{N}(L)}(h_0^{(0)}) = h_0^{(0)} = \left(C\big|_{\mathcal{N}(L)}\right)^\dagger (b^\delta - Ax_0).$$

In force of Assumption 6.2(i) we have

$$P_{\mathcal{N}(L)}(h_0) = \left(A\big|_{\mathcal{N}(L)}\right)^\dagger (b^\delta - Ax_0) = P_{\mathcal{N}(L)}(A^\dagger b^\delta) - P_{\mathcal{N}(L)}(x_0).$$

And thus

$$P_{\mathcal{N}(L)}(x_1) = P_{\mathcal{N}(L)}(x_0) + P_{\mathcal{N}(L)}(A^\dagger b^\delta) - P_{\mathcal{N}(L)}(x_0) = P_{\mathcal{N}(L)}(A^\dagger b^\delta),$$

so in the null space of L we directly invert the operator A at the very first step.

Proposition 6.12. *Let $\overline{e}_k = Le_k^\perp$, under Assumption 6.2 the norm of $\overline{e}_k$ of Algorithm 6.2 decreases monotonically.*

$$\|\overline{e}_k\|^2 - \|\overline{e}_{k+1}\|^2 \geq 2\rho \left\|(\overline{CC}^* + \alpha_k I)^{-1}\overline{r}_k\right\| \|r_k\|$$

Proof. This proof is in the spirit of the original result showed in [49, Proposition 2]. Let us consider $\|\overline{e}_k\| = \left\|Le_{k+1}^\perp\right\| = \|Le_{k+1}\|$

$$\begin{aligned}\|\overline{e}_{k+1}\|^2 &= \langle Le_{k+1}, Le_{k+1}\rangle = \langle Le_k - Lh_k, Le_k - Lh_k\rangle \\ &= \|Le_k\|^2 - 2\langle Le_k, Lh_k\rangle + \|Lh_k\|^2.\end{aligned}$$

Using the definition of h_k, denoting with $Q_k = (\overline{CC}^* + \alpha_k I)$, it holds

$$\begin{aligned}\|\overline{e}_k\|^2 - \|\overline{e}_{k+1}\|^2 &= 2\langle Le_k, Lh_k\rangle - \|Lh_k\|^2 \\ &\geq 2\langle Le_k, Lh_k\rangle - 2\|Lh_k\|^2 \\ &= 2\left\langle Le_k, \overline{C}^* Q_k^{-1}\overline{r}_k\right\rangle - 2\left\langle \overline{r}_k, \overline{CC}^* Q_k^{-2}\overline{r}_k\right\rangle.\end{aligned}$$

since $Lh_k = \overline{h}_k = \overline{C}^* Q_k^{-1} \overline{r}_k$ thanks to (6.6). Therefore

$$\begin{aligned}
\|\overline{e}_k\|^2 - \|\overline{e}_{k+1}\|^2 &= 2\left\langle \overline{r}_k, Q_k^{-1}\overline{r}_k \right\rangle - 2\left\langle \overline{r}_k, \overline{CC}^* Q_k^{-2}\overline{r}_k \right\rangle \\
&\quad - 2\left\langle \overline{r}_k - \overline{C}Le_k, Q_k^{-1}\overline{r}_k \right\rangle \\
&= 2\left\langle \overline{r_k}, Q_k^{-1}\overline{r}_k \right\rangle - 2\left\langle \overline{r}_k, \overline{CC}^* Q_k^{-2}\overline{r}_k \right\rangle \\
&\quad - 2\left\langle r_k - Ce_k^{\perp}, Q_k^{-1}\overline{r}_k \right\rangle \\
&= 2\left\langle \overline{r}_k, \left[Q_k^{-1} - \overline{C}C^* Q_k^{-2}\right]\overline{r}_k \right\rangle - 2\left\langle \overline{r}_k - Ce_k^{\perp}, Q_k^{-1}\overline{r}_k \right\rangle \\
&\geq 2\alpha_k \left\langle \overline{r}_k, Q_k^{-2}\overline{r}_k \right\rangle - 2\left\| \overline{r}_k - Ce_k^{\perp} \right\| \left\| Q_k^{-1}\overline{r}_k \right\| \\
&= 2\alpha_k \left\| Q_k^{-1}\overline{r}_k \right\|^2 - 2\left\| \overline{r}_k - Ce_k^{\perp} \right\| \left\| Q_k^{-1}\overline{r}_k \right\| \\
&= 2\left\| Q_k^{-1}\overline{r}_k \right\| \left[\left\| \alpha_k Q_k^{-1}\overline{r}_k \right\| - \left\| \overline{r}_k - Ce_k^{\perp} \right\| \right] \\
&\geq 2\left\| Q_k^{-1}\overline{r}_k \right\| \left[\|r_k - Ch_k\| - \|r_k - Ce_k\| \right],
\end{aligned}$$

where the last step is obtained by considering (6.7)

$$\begin{aligned}
\|r_k - Ch_k\| &= \left\| \overline{r}_k - \overline{Ch}_k \right\| = \left\| \overline{r}_k - \overline{CC}^* (\overline{CC} * + \alpha_k I)^{-1} \overline{r}_k \right\| \\
&= \left\| \left[I - \overline{CC}^* (\overline{CC} * + \alpha_k I)^{-1} \right] \overline{r}_k \right\| = \left\| \alpha_k Q_k^{-1} \overline{r}_k \right\|,
\end{aligned}$$

and by

$$\left\| \overline{r}_k - Ce_k^{\perp} \right\| = \left\| P_{\mathcal{N}(L)^{\perp}} (r_k - Ce_k) \right\| \leq \|r_k - Ce_k\|,$$

since $\left\| P_{\mathcal{N}(L)^{\perp}} \right\| = \left\| L^{\dagger} L \right\| = 1$.

In virtue of Proposition 6.1 and using equation (6.4) we have that

$$\begin{aligned}
\|\overline{e}_k\|^2 - \|\overline{e}_{k+1}\|^2 &\geq 2\left\| Q_k^{-1}\overline{r}_k \right\| \left[q_k \|r_k\| - \|r_k - Ce_k\| \right] \\
&\geq 2\rho \left\| Q_k^{-1}\overline{r}_k \right\| \|r_k\| = 2\rho \left\| (\overline{CC}^* + \alpha_k I)^{-1} \overline{r}_k \right\| \|r_k\|.
\end{aligned}$$

□

We call k^{δ} the iteration at which Algorithm 6.2 stops. From Corollary 6.13 we are going to be able to deduce that k^{δ} is finite if $\delta > 0$, independently of the choice of $\mathbf{x}_0$.

Repeating the same steps that in [49] led to derive Corollary 3 from Proposition 2, the following result can be derived from Proposition 6.12.

Corollary 6.13. *With the notation and assumptions of Proposition 6.12, it holds*

$$\|\overline{e}_0\|^2 \geq 2\rho \sum_{k=0}^{k^{\delta}-1} \left\| (\overline{CC}^* + \alpha_k I)^{-1} r_k \right\| \|r_k\| \geq c \sum_{k=0}^{k^{\delta}-1} \|r_k\|^2.$$

Form the outer inequality in Corollary 6.13 we obtain that the sum of the squares of the norm of the residual (in $\mathcal{N}(L)^{\perp}$) is bounded and hence, if $\delta > 0$, there must be a first integer $\mathbb{N} \ni k^{\delta} < \infty$ that fulfills the stopping criterion. In fact suppose that the algorithm does not stop after finitely many iterations, we get that $\lim_{k\to\infty} \|r_k\|^2 = 0$. Thus there exists $\overline{k}$ such that $\left\| r_{\overline{k}} \right\| < \tau\delta$ which is absurd. In other words, if $\delta > 0$ Algorithm 6.2 terminates after a

finite number of iterations. Conversely, in Theorem 6.14 we show that, if $\delta = 0$ and x_0 is not a solution of the system, then the algorithm, even though it converges to a solution of the system, does not stop.

Theorem 6.14. *Assume that the data are exact, i.e., $\delta = 0$, and that x_0 is not a solution of the problem. Then, the sequence $(x_k)_k$ converges as $k \to \infty$ to a solution $x_0^\dagger$ such that:*

(i) $Ax_0^\dagger = b$;

(ii) $P_{\mathcal{N}(L)}(x_0^\dagger) = P_{\mathcal{N}(L)}(x^\dagger)$;

(iii) the distance between $x_0^\dagger$ and x_0 is minimal with respect to the set of all the solutions.

Proof. The proof follows the same strategy of the analogous result in [49, Theorem 4]. Let us call $x_{k^\delta} = x_0^\dagger$, since $\delta = 0$ the stopping criterion can only be fulfilled for $k = k^\delta$ and with $\|r_k\| = 0$.

We now show that an infinite number of iterations is needed. If $k > 0$, then h_{k-1} must coincide with e_{k-1} up to an element in the null space of A, that is (thanks to Assumption 6.1) the null space of C, and so, using (6.4) and Proposition 6.1, we get

$$q_{k-1} \|r_{k-1}\| = \|r_{k-1} - Ch_{k-1}\| = \|r_{k-1} - Ce_{k-1}\| \leq \left(\rho + \frac{1+\rho}{\tau_{k-1}}\right) \|r_{k-1}\| .$$

This contradicts the definition of q_{k-1} and so the iteration does not terminate after finitely manly iterations for exact data if x_0 is not a solution of the system.

Using Remark 6.11, the proof of point (ii) is immediate. It is left for us to show points (iii) and (i). We first show that the sequence $(Lx_k)_k = (\overline{x}_k)_k$ is a Cauchy sequence.

Let $j > l$ and let us consider $\|Lx_j - Lx_l\|^2$

$$\begin{aligned} \|Lx_j - Lx_l\|^2 = \|Le_j - Le_l\|^2 &= \|\overline{e}_j\|^2 - \|\overline{e}_l\|^2 - 2\langle \overline{e}_l, \overline{e}_j - \overline{e}_l \rangle \\ &= \|\overline{e}_j\|^2 - \|\overline{e}_l\|^2 + 2\langle \overline{e}_l, \overline{x}_j - \overline{x}_l \rangle . \end{aligned}$$

Inserting the definition of x_k and of h_k we get

$$\begin{aligned} \|\overline{x}_j - \overline{x}_l\|^2 &= \|\overline{e}_j\|^2 - \|\overline{e}_l\|^2 + 2\sum_{i=l}^{j-1} \langle \overline{e}_l, \overline{h}_i \rangle \\ &= \|\overline{e}_j\|^2 - \|\overline{e}_l\|^2 + 2\sum_{i=l}^{j-1} \left\langle Le_l, \overline{C}^*(\overline{CC}^* + \alpha_i I)^{-1}\overline{r}_i \right\rangle \\ &= \|\overline{e}_j\|^2 - \|\overline{e}_l\|^2 + 2\sum_{i=l}^{j-1} \left\langle Ce_l^\perp, (\overline{CC}^* + \alpha_i I)^{-1}\overline{r}_i \right\rangle \\ &\leq \|\overline{e}_j\|^2 - \|\overline{e}_l\|^2 + 2\sum_{i=l}^{j-1} \left\|Ce_l^\perp\right\| \left\|(\overline{CC}^* + \alpha_i I)^{-1}\overline{r}_i\right\| \\ &\leq \|\overline{e}_j\|^2 - \|\overline{e}_l\|^2 + 2\sum_{i=l}^{j-1} \|Ce_l\| \left\|(\overline{CC}^* + \alpha_i I)^{-1}\overline{r}_i\right\|, \end{aligned}$$

where in the last step we have used the fact that $\left\|Ce_i^\perp\right\| \leq \|Ce_i\|$.

Let us suppose now that $l \geq k$ and so

$$\begin{aligned}
\|\overline{x}_l - \overline{x}_k\|^2 &= \|\overline{e}_k\|^2 - \|\overline{e}_l\|^2 + 2\sum_{i=k}^{l-1} \langle \overline{e}_l, \overline{h}_i \rangle \\
&= \|\overline{e}_k\|^2 - \|\overline{e}_l\|^2 + 2\sum_{i=k}^{l-1} \left\langle Ce_l^{\perp}, (\overline{CC}^* + \alpha_i I)^{-1}\overline{r}_i \right\rangle \\
&\leq \|\overline{e}_k\|^2 - \|\overline{e}_l\|^2 + 2\sum_{i=k}^{l-1} \left\| Ce_l^{\perp} \right\| \left\| (\overline{CC}^* + \alpha_i I)^{-1}\overline{r}_i \right\| \\
&\leq \|\overline{e}_k\|^2 - \|\overline{e}_l\|^2 + 2\sum_{i=k}^{l-1} \|Ce_l\| \left\| (\overline{CC}^* + \alpha_i I)^{-1}\overline{r}_i \right\|.
\end{aligned}$$

Using the two inequalities together and Assumption 6.1 we get for general $j > k$ and any $l \in \{k, \dots, j-1\}$

$$\begin{aligned}
\|Lx_j - Lx_k\|^2 &\leq 2\|Lx_j - Lx_l\|^2 - 2\|Lx_l - Lx_k\|^2 \\
&\leq 2\|\overline{e}_j\|^2 + 2\|\overline{e}_k\|^2 + \\
&\quad - 4\|\overline{e}_l\|^2 + 4\sum_{i=k}^{j-1} \|Ce_l\| \left\| (\overline{CC}^* + \alpha_i I)^{-1}\overline{r}_i \right\| \\
&\leq 2\|\overline{e}_j\|^2 + 2\|\overline{e}_k\|^2 - 4\|\overline{e}_l\|^2 + \\
&\quad + 4(1+\rho)\sum_{i=k}^{j-1} \|r_l\| \left\| (\overline{CC}^* + \alpha_I I)^{-1}\overline{r}_i \right\|
\end{aligned}$$

Let $l \in \{k, \dots, j-1\}$ be that particular index for which $\|r_l\|$ is minimal, so that

$$\begin{aligned}
\|Lx_j - Lx_k\|^2 \leq & 2\|\overline{e}_j\|^2 + 2\|\overline{e}_k\|^2 - 4\|\overline{e}_l\|^2 + \\
& + 4(1+\rho)\sum_{i=k}^{j-1} \|r_i\| \left\| (\overline{CC}^* + \alpha_i I)^{-1}\overline{r}_i \right\|.
\end{aligned}$$

The right-hand side of the inequality above becomes arbitrarily small, because the sequence $(\|\overline{e}_k\|)_k$ is monotonically decreasing, in force of Proposition 6.12, and so it converges to some limit $\epsilon \geq 0$ and the summation is the partial sum of a converging series (see Corollary 6.13). We have proved that the sequence $(\overline{x}_k)_k$ is a Cauchy sequence and so it converges to a certain limit $\overline{x} \in \mathcal{X}$. By continuity of $L^\dagger$ we get that

$$P_{\mathcal{N}(L)^\perp}(x_k) = L^\dagger L x_k \to L^\dagger \overline{x} = L^\dagger L x = P_{\mathcal{N}(L)^\perp}(x),$$

for some $x \in \mathcal{X}$. Accordingly the norm of the residual $P_{\mathcal{N}(L)^\perp}(r_k) = P_{\mathcal{N}(L)^\perp}(b - Ax_k)$ goes to $P_{\mathcal{N}(L)^\perp}(b - Ax)$, while in force of Corollary 6.13 the norm of this residual converges to zero and so $P_{\mathcal{N}(L)^\perp}(x)$ is the projection of a solution of the system, this with Remark 6.11 proves point (i) of the theorem.

By construction, every iterate x_k satisfies

$$x_k - x_0 = \sum_{k=0}^{n-1} h_k \in \mathcal{R}(C^*) = \mathcal{N}(C)^\perp,$$

Therefore $x - x_0 \in \mathcal{N}(A)^\perp$, thanks to Assumption 6.2 (i) and so x is the particular solution of the system which is closest to x_0 in the norm of $\mathcal{X}$ thus proving point (iii). □

Remark 6.15. *If x_0 is a solution of the system then we have that $\|r_0\| = \|Ax_0 - b\| = 0$ and thus the algorithm does not start. In particular only a finite number of iteration is needed.*

Let us consider the inexact data case, in this circumstances Algorithm 6.2 is a regularization method, in fact we have the following

Theorem 6.16. *Assume that Assumption 6.2 holds for some $0 < \rho \leq \frac{1}{2}$ and let $\delta \mapsto b^\delta$ be a function from $\mathbb{R}$ to $\mathcal{X}$ such that for all δ it holds $\left\|b - b^\delta\right\| \leq \delta$. For fixed τ and q denote by x^δ the approximation of $x^\dagger$ obtained with Algorithm 6.2. Then, as $\delta \to 0$, x^δ goes to the solution of the system which is closest to x_0.*

We omit the proof since it can be copied from [77, Theorem 2.3]; for further reference see also [61, Theorem 11.5]. Its essentials ingredients are the monotonicity proved in Proposition 6.12, the convergence to the exact solution in the exact data case proved in Theorem 6.14 and the continuity of the map $\delta \mapsto b^\delta$.

6.3 Approximated Projected Iterated Tikhonov (APIT)

Let $\Omega \subset \mathcal{X}$ be closed and convex and such that $x^\dagger \in \Omega$, let P_Ω be the metric projection of $\mathcal{X}$ on Ω and $A_\Omega = A\big|_\Omega$, $C_\Omega = C\big|_\Omega$. We want to constrain our problem so that $\forall k,\ x_k \in \Omega$.

Definition 6.17. *We define the* metric projection *of $x \in \mathcal{X}$ onto Ω as*

$$P_\Omega(x) = \arg\min_{y \in \Omega} \frac{1}{2} \|x - y\|^2 .$$

Lemma 6.18. *Let Ω be a closed and convex subset of a Hilbert space $\mathcal{X}$, then P_Ω, the metric projection of $\mathcal{X}$ over n, is such that:*

(i) $\|P_\Omega(x) - P_\Omega(y)\|^2 \leq \|x - y\|^2 - \|(I - P_\Omega)(x) - (I - P_\Omega)(y)\|^2$;

(ii) $\|P_\Omega(x) - P_\Omega(y)\|^2 \leq \langle x - y, P_\Omega(x) - P_\Omega(y)\rangle$.

Proof. The proof of the first can be found in [121]. The second is just a reformulation. □

Remark 6.19. *Lemma 6.18 implies that the map P_Ω is* non-expansive.

In order to constrain Algorithm 6.1 we simply project at each iteration, obtaining the following

Algorithm 6.3 (Approximated Projected Iterated Tikhonov (APIT))**.** *Let $x_0 \in \mathcal{X}$ be fixed. Choose $\tau = \frac{1+2\rho}{1-2\rho}$ with ρ as in* (6.1)*, and fix $q \in [2\rho, 1]$.*

$$
\begin{aligned}
&k = 0,\ r_0 = b^\delta - Ax_0,\ \tau_0 = \frac{\|r_k\|}{\delta} \\
&\texttt{While}\ \ \|r_k\| > \tau\delta \\
&\quad \tau_k = \frac{\|r_k\|}{\delta} \\
&\quad q_k = \max\left\{q, 2\rho + \frac{1+\rho}{\tau_k}\right\} \\
&\quad \texttt{compute}\ \alpha_k\ \texttt{such that}\ \ \left\|r_k - CC^*(CC^* + \alpha_k I)^{-1} r_k\right\| = q_k \|r_k\| \\
&\quad h_k = C^*(CC^* + \alpha_k I)^{-1} r_k \\
&\quad x_{k+1} = P_\Omega(x_k + h_k) \\
&\quad r_{k+1} = b^\delta - Ax_{k+1}
\end{aligned}
$$

We refer to Algorithm 6.3 as *Approximated Projected iterated Tikhonov* since this method can be seen as a preconditioned iterative method whose preconditioner is obtained by approximated Tikhonov and is projected at each iteration.

Remark 6.20. *Since $x^\dagger \in \Omega$, we have $\|e_k\| \leq \|\tilde{e}_k\|$, where $\tilde{e}_k$ is the error at the n-th iteration before of the projection into Ω, namely $\tilde{e}_k = x^\dagger - (x_{k-1} + h_{k-1})$.*

Using Lemma 6.18, the theoretical results reported in Section 6.1 for AIT can be easily extended to APIT.

Proposition 6.21. *With the same notations and assumptions of Proposition 6.2, the norm of the iteration error e_k decreases monotonically, namely*

$$\|e_k\|^2 - \|e_{k+1}\|^2 \geq 2\rho \left\|(CC^* + \alpha_k I)^{-1} r_k\right\| \|r_k\| .$$

Proof. Using Lemma 6.18:

$$
\begin{aligned}
\|e_k\|^2 - \|e_{k+1}\|^2 = \|e_k\|^2 - \left\|x^\dagger - P_\Omega(x_k + h_k)\right\|^2 \geq \\
\geq \|e_k\|^2 - \left\|x^\dagger - (x_k + h_k)\right\|^2 = \|e_k\|^2 - \|e_k - h_k\|^2 .
\end{aligned}
$$

Then, proceeding like in [49, Proposition 2], we have the thesis. □

Using the same approach of [49] to prove the results in Section 6.2, it can be shown that[1]

Theorem 6.22. *Assume that the data are correct, i.e., $\delta = 0$, and that x_0 is not a solution of the problem* (2.3)*. Then, the sequence x_k converges as $n \to \infty$ to a solution of* (2.3)*.*

Using this result and copying the proof of Theorem 2.3 in [77] we obtain

Theorem 6.23. *Let $\delta \mapsto b^\delta$ be a function from $\mathbb{R}^+$ to $\mathcal{Y}$ such that* (2.2) *holds true for all $\delta > 0$. Under Assumption 6.1, for fixed parameters τ and q, denote by n_δ the corresponding stopping indexes, and by x^δ the resulting approximations. Then, as $\delta \to 0$, x^δ converges to a solution of* (2.3)*.*

[1] For more details see the proof of Theorem 7.10 in Chapter 7

6.3.1 Approximated Projected Iterated Tikhonov with General Penalty term (APIT-GP)

We now combine the previous two algorithms into a third one.
Algorithm 6.4 (Approximated Projected Iterated Tikhonov with General Penalty term (APIT-GP))**.** *Let $x_0 \in \mathcal{X}$ be fixed, set $k = 0$. Choose $\tau = \frac{1+2\rho}{1-2\rho}$ with ρ as in* (6.1)*, and fix $q \in [2\rho, 1]$.*

$$
\begin{aligned}
&k = 0,\ r_0 = b^\delta - Ax_0,\ \tau_0 = \frac{\|r_k\|}{\delta} \\
&\texttt{While}\ \ \|r_k\| > \tau\delta \\
&\quad \tau_k = \frac{\|r_k\|}{\delta} \\
&\quad q_k = \max\left\{q, 2\rho + \frac{1+\rho}{\tau_k}\right\} \\
&\quad \texttt{compute}\ \alpha_k\ \texttt{such that}\ \ \left\|r_k - CC^*(CC^* + \alpha_k LL^*)^{-1} r_k\right\| = q_k \|r_k\| \\
&\quad h_k = C^*(CC^* + \alpha_k LL^*)^{-1} r_k \\
&\quad x_{k+1} = P_\Omega(x_k + h_k) \\
&\quad r_{k+1} = b^\delta - Ax_{k+1}
\end{aligned}
$$

We refer to Algorithm 6.4 as *Approximated Projected Iterated Tikhonov with General Penalty term* since this method can be seen as the combination of Algorithms 6.2 and 6.3.

From the numerical experiments in Section 6.4 we can see that this algorithm has good performances, however, we are not able to provide a proof of its convergence.

6.4 Numerical Examples

We apply our methods to the image deblurring problem. In this examples we use as A the blurring matrix with boundary conditions that respect the nature of the image and C the blurring matrix that has the same point PSF of A with periodic boundary conditions.

As L we choose the divergence with periodic boundary conditions defined in (2.14). Recall that L is singular and its null space is

$$\mathcal{N}(L) = \operatorname{span}\{\mathbf{1}\}. \tag{6.8}$$

Since we are using the periodic boundary conditions the C and L are BCCB matrices and satisfy the Assumption 6.2 (ii), with F^* being the two-dimensional discrete Fourier transform [84].

Note that for L defined in (2.14), we have that $\mathcal{N}(L) \cap \mathcal{N}(C) = \{\mathbf{0}\}$ thanks to (6.8). Indeed,

$$C\mathbf{1} = \mathbf{1}$$

because the sum of every row of C is equal to the sum of all entries of the PSF, which is equal to 1 to preserve the total light intensity.

Images can be seen as the measurement of the quantity of light received from a source and so they should not have negative values. Therefore we choose Ω to be the nonnegative cone defined in (3.2).

Moreover, according to several numerical tests with different problems and the suggestions in [49], we fix

$$\rho = 10^{-3} \quad \text{and} \quad q = 0.7$$

in all our examples. To compare the quality of the restorations, we use the RRE defined in (2.15). The minimum RRE in all table will be marked in bold.

For the construction of the examples we proceed in the following way. We first start with an image of $n_1 \times n_2$ pixels and blur it using any boundary conditions, e.g., the periodic one, using a PSF with of $m_1 \times m_2$ with $m_j < n_j,\ j = 1, 2$. Then, in order to simulate a real situation, we cut out the the boundary from the blurred image of half the size of the PSF, i.e., of $\lceil \frac{m_j}{2} \rceil$. We then add some white Gaussian noise with noise level ξ defined in (2.13).

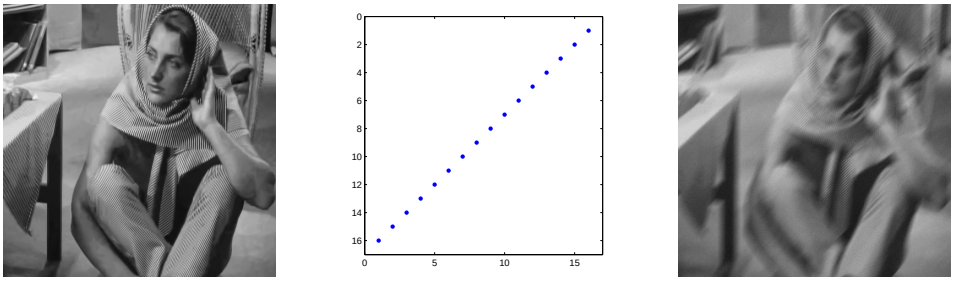
We compare the restoration obtained with our methods with the original method AIT and with some other methods already present in the literature. In particular we consider the following methods:

- Hybrid [40];
- Two step iterative shrinkage/thresholding (Twist) [15];
- Range Restricted Arnoldi–Tikhonov (RRAT) [104];
- Flexible Arnoldi Tikhonov(FlexiAT) [67];
- Nonnegative Restarted Generalized Arnoldi Tikhonov (NN-ReStart-GAT) [67].

The Hybrid method is a Krylov method in which on each Krylov space a Tikhonov regularization is implemented so to obtain a regularized solution, the regularization parameter is chosen with a particular modification of the generalized cross validation. In RRAT the Arnoldi Tikhonov decomposition is used to consider a certain Krylov space and then on this space the regularized solution is obtained using Tikhonov regularization, the regularization parameter is chosen solving the discrepancy principle equation. Twist is a method that combines regularization of the iterative shrinkage/thresholding methods and the splitting of the Iterative Re-Weighted Shrinkage methods. FlexiAT is a method that enables to introduce a regularization term into the equation and to adapt the Krylov subspace using the intermediate solutions in order to achieve better approximates the optimal regularization matrix. NN-ReStart-GAT is a projected version of ReStart-GAT, this method uses a restarted strategy, the inner iteration solves a Tikhonov regularized version of the problem exploiting Arnoldi Tikhonov decomposition with a regularization term L such that $\|Lx\| \approx \|x\|_1$, then the outer iteration updates L so that the approximation of the $1-$norm gets better and better with the iterations.

In the following figures, the restored images are shown after a projection into Ω also for the methods that do not impose the nonnegative constraint. This allows a better visualization of the images in particular when they are affected by large ringing effects.

All the tests were performed using MATLAB 9.0.0.341360 (R2016a) 64bit running on a laptop with an Intel core i7-6700HQ @ 2.60 GHz CPU and 8 GB of RAM.



(a) (b) (c)

FIGURE 6.1: Barbara test problem: (a) Test image (496×496 pixels), (b) Diagonal motion PSF (16×16 pixels), (c) Blurred image (496×496 pixels), $RRE = 0.16145$.

(a) (b) (c) (d)

FIGURE 6.2: Barbara test problem reconstructions: (a) AIT-GP, (b) AIT (c) Hybrid at the optimal iteration (the method does not stop properly), (d) TwIST.

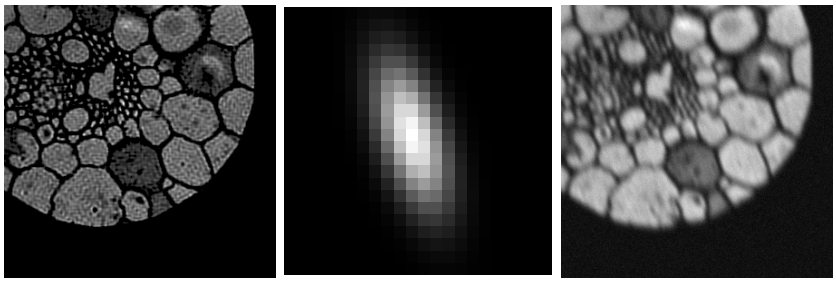
Barbara In this example we use the image Barbara, we blur the image with a diagonal motion PSF of 16 pixel and add 3% of white Gaussian noise, i.e., $\xi = 0.03$ in (2.13), see Figure 6.1.

Since the image is generic we use the antireflective boundary conditions for the operator A. From the comparison of the RRE history in Figure 6.8(a) we can see that, since there are no important black parts in the image, the introduction of the projection does not give any relevant improvement, in fact the graphs of APIT and AIT are overlayed and the same happens for AIT-GP and APIT-GP. The introduction of L is able to make the method faster and more accurate. In Figure 6.2 we can see the reconstructions with AIT-GP, AIT, Twist and the optimal reconstruction obtained with Hybrid. We can see from those reconstructions that the introduction of the regularization operator L let us have a better reconstruction of edges and details even though there are some ringing effects. In Table 6.1 we can find the comparison of the RRE and computational times with some other method from the literature, we can see that usually the method proposed are able to get better reconstructions in a smaller amount of time. We want to stress the fact that the stopping criterion of Hybrid was not able to effectively stop the method, so we printed also the optimal error; from this we can see that, even though AIT and APIT are outperformed by Hybrid, the introduction of the regularization operator gives better reconstructions.

FlexiAT, RRAT and NN-ReStart-GAT do not seems to perform well, moreover they also reach the maximum number of iterations without converging. This effect might be due to the fact that this methods are constructed for images that are mostly black, like astronomical or biological images, and not for photographic images like the one we are using in this example.

Method	RRE	Iterations	Computational Time (sec.)
AIT	0.13489	3	0.60364
AIT-GP	**0.13132**	3	0.75130
APIT	0.13489	3	0.57012
APIT-GP	**0.13132**	3	0.73076
Hybrid	0.15919 (Opt.: 0.13337)	33 (Opt.: 5)	9.4639 (Opt. 1.4339)
TwIST	0.13906	6	4.5313
FlexiAT	0.16613	50	6.6665
RRAT	0.17308	50	13.782
NN-ReStart-GAT	0.16471	500	109.10

TABLE 6.1: Barbara test problem: Comparison of the methods in term of RRE, number of iterations and computational time. The smallest error is shown in boldface.

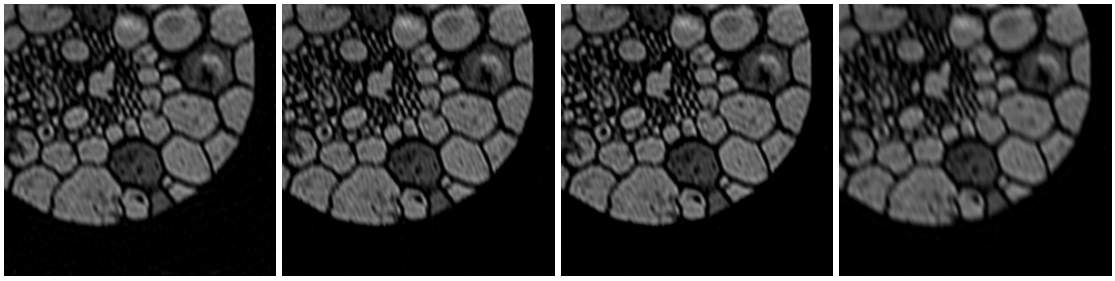


(a) (b) (c)

FIGURE 6.3: Grain test problem: (a) Test image (300×300 pixels), (b) Non symmetric Gaussian PSF (22×22 px), (c) Blurred image (300×300 pixels), $RRE = 0.3680$.

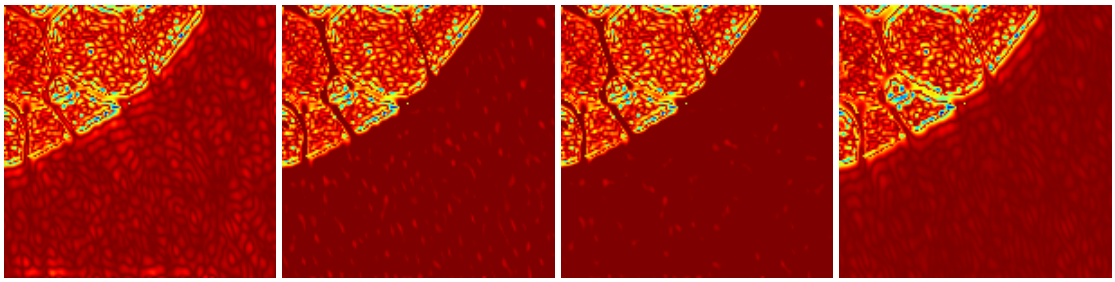
Grain For this example we use the image Grain, we blur the image with a non symmetric Gaussian PSF from the toolbox RESTORE TOOLS[12], and add 5% of white Gaussian noise, i.e., $\xi = 0.05$ in (2.13) (for the result see Figure 6.3). This image having a very huge black area is very useful to see the improvements introduced by the nonnegative constraint. Again, since the image is generic at the boundary, we use the antireflective boundary conditions. From the RRE history in Figure 6.8(b) we can see that the projection in the nonnegative cone gives great improvements in the quality of the reconstructions. In Figure 6.4 we can find the reconstruction with the AIT-GP, APIT, APIT-GP and Hybrid methods, from these we can see that L helps reconstructing the edges and that the projection let us have a more homogeneous result in the black areas. In order to better notice that we show in Figure 6.5 a detail of $|e_{n^\delta}|$ in color map jet. In fact the reconstructions in Figure 6.4 are visualized so that no negative values are introduced, if the negative values were permitted we would get high oscillations in the black areas for the non-projected algorithms. In Table 6.2 we can find the RRE and computational times of our algorithms compared with some other method from the literature.

Satellite In this last example we use the dataset `satellite` from the toolbox RESTORE TOOLS[12]. In this case the image is blurred with an astronomical PSF. The noise level ξ is approximately of the 4% and has been computed using the knowledge of the true image. See Figure 6.6 for the true image, the PSF and the blurred and noisy data. Like in the example before this image, having a very huge black area, is very useful to see the improvements introduced by the nonnegative constraint. Since near the boundary the image is all black we use the *zero* boundary conditions. In Figure 6.8(c) we find the RRE history, we can see



(a) (b) (c) (d)

FIGURE 6.4: Grain test problem reconstructions: (a) AIT-GP, (b) APIT, (c) APIT-GP, (d) Hybrid.



(a) (b) (c) (d)

FIGURE 6.5: Grain test problem, absolute value of the error in the south-east corner: (a) AIT-GP, (b) APIT, (c) APIT-GP, (d) Hybrid.

Method	RRE	Iterations	Computational Time (sec.)
AIT	0.28742	4	0.33530
AIT-GP	0.28485	4	0.33922
APIT	0.27393	30	1.4502
APIT-GP	**0.27063**	57	3.0685
Hybrid	0.32334	8	1.0395
TwIST	0.28743	16	5.3594
FlexiAT	0.35340	4	2.7535
RRAT	0.29767	9	0.084399
NN-ReStart-GAT	0.35044	52	4.7067

TABLE 6.2: Grain test problem: Comparison of the methods in term of RRE, number of iterations and computational time. The smallest error is shown in boldface.

FIGURE 6.6: Satellite test problem: (a) Test image (256×256 pixels), (b) Astronomic PSF (256×256 pixels), (c) Blurred image (256×256 pixels), $RRE = 0.70464$.

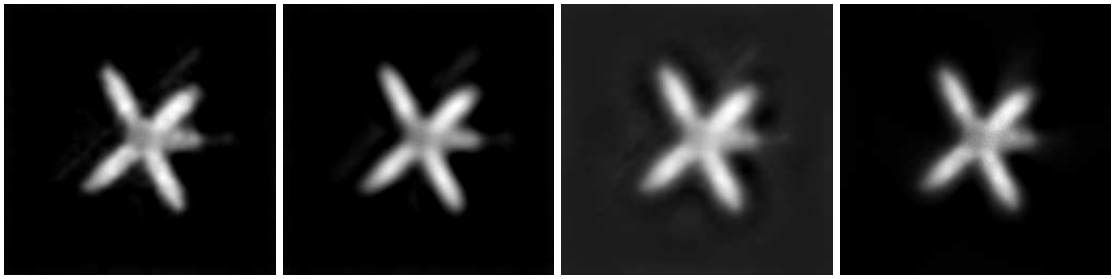
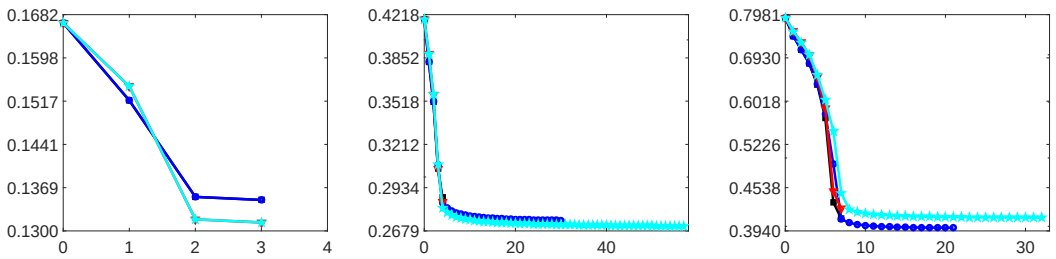


FIGURE 6.7: Satellite test problem reconstructions: (a) APIT, (b) APIT-GP, (c) RRAT, (d) FlexiAT.

Method	RRE	Iterations	Computational Time (sec.)
AIT	0.40996	7	0.48016
AIT-GP	0.42385	7	0.50797
APIT	**0.39801**	21	1.1806
APIT-GP	0.41129	32	1.7659
Hybrid	0.47663	50	4.5397
TwIST	0.47745	22	3.0313
FlexiAT	0.44875	8	0.18139
RRAT	0.45807	8	0.078776
NN-ReStart-GAT	0.83804	59	4.5397

TABLE 6.3: Satellite test problem: Comparison of the methods in term of RRE, number of iterations and computational time. The smallest error is shown in boldface.

that all the three methods we introduced give better result than AIT, since the image is for the most part black the better result is achieved with APIT. In this case, however, the best reconstruction is not the one given by APIT-GP, this is due to the fact that the introduction of the regularization operator is able to enhance the edges and some small noise, in the black area is recognized as edge and preserved. We must notice, none the less, that the difference between APIT-GP and APIT is very small. Finally in Figure 6.7 we can see the reconstructions for APIT, APIT-GP, RRAT and NN-Restart-GAT. In Table 6.3 we can find the comparison of the RRE and computational times with some other method from the literature, we can see that all the method proposed are able to get better reconstructions, even though in some cases the computational time is higher.



(a) (b) (c)

FIGURE 6.8: Evolution of the relative reconstruction error against the iterations for AIT, APIT, AIT-GP, and APIT-GP: (a) Barbara test case, (b) Cell test case, (c) Satellite test case. In black with stars AIT, in blue with circles APIT, in red with triangles AIT-GP, and in cyan with pentacles APIT-GP.

Chapter 7

Multigrid iterative regularization method for image deblurring with arbitrary boundary conditions

In this chapter we turn again our attention to the finite dimensional case of the linear system of equations (2.3). We are going to discuss a multigrid algorithm designed for image deblurring and denoising. In particular, we want to construct an iterative regularization method using the multigrid framework.

Multigrid methods are very powerful algorithms that are able to achieve very fast computations and high accuracy, see e.g. [24, 120]. Multigrid methods have been initially developed for solving linear systems of equations derived from partial differential equations (PDEs) [22] and later successfully applied to more general linear systems [118].

Multigrid methods have already been considered to solve ill-posed problems [36, 37, 55, 81, 92, 94, 114], but usually as solvers for Tikhonov like regularized models. The first attempt of using multigrid methods as iterative regularization methods has been probably done in [56], where the authors combined an iterative regularization method used as pre-smoother with a low-pass filter coarsening. Later the same authors furtherly discussed in [55] the regularizing properties of this method. A different multilevel strategy based on the cascadic approach was proposed in [110]. Nonlinear "corrections" to the previous multigrid methods were introduced in [100] using a total variation-type regularization and in [45, 63] combining multigrid and wavelets. More recently, also the blind deconvolution has been successfully approached in [62]. Note that, these multigrid methods have been defined to preserve the BTTB structure of the blurring matrix at each coarser level. This is crucial for the definition of the algorithm and for preserving a fast and simple matrix vector product.

The main novelty in [45], with respect to [56], was the addition of a soft-thresholding denoising as post-smoother. We are going to define our method by starting from the idea in [45] of combining framelet denoising and multigrid. Firstly, differently from the previous works, we define a general coarsening strategy independently of the boundary conditions. In practice, the Galerkin projection of the operator is applied to the PSF instead of to the coefficient matrix and at each coarser level we can apply the favorite boundary conditions. Furthermore, our proposal differs from the one in [45] also for the use of framelet denoising as a pre-smoother instead of as post-smoother and for the use of APIT, described in Chapter 6, instead of CGLS as inner iterative regularization method. This choice let us ensure the nonnegativity of the provided approximation, because APIT projects each iteration into the nonnegative cone and so using it as post-smoother it is equivalent to project each multigrid iteration into the nonnegative cone. Finally we also give a theoretical proof of convergence

of the algorithm in the two grid case (as usual for multigrid methods [120]) under some restrictive, but reasonable, hypothesis.

Using the knowledge of δ the proposed algorithm is able to achieve very high accuracy without tuning any parameter.

This chapter is structured as follows: in Section 7.1 we briefly describe the multigrid algorithm and the framelet denoising, which are both needed for the formulation of our algorithm, in Section 7.2 we describe our algorithmic proposal, in Section 7.3 we discuss the convergence of the method and, finally, in Section 7.4 we give some numerical examples.

7.1 Preliminaries

In this section we present some tools which are needed in the construction of our algorithm.

7.1.1 Multigrid Methods

The first tool which we have to describe is the Multigrid approach.

Multigrid methods have been developed for solving partial differential equation and, more in general, for solving linear systems of equations of large size. The basic idea of the multigrid is to create a sequence of linear systems which get smaller and smaller by consecutive projection. In this way the computational effort can be reduced and the convergence speed can be improved up if the smaller linear systems are properly chosen.

Let us start with the linear system of equation

$$A\mathbf{x} = \mathbf{b}, \tag{7.1}$$

where $A \in \mathbb{R}^{n\times n}$ is invertible and $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$.

It is well known that iterative methods first converge in the well-conditioned space and that the convergence can be very slow in the ill-conditioned space. Differently, while direct methods are not affected by this kind of problem, they are usually much more expensive and are more sensible to error propagation.

Remark 7.1. *The definition of well- and ill-conditioned space is not formal.*

Let $V \subset \mathbb{R}^n$ be a linear subspace of $\mathbb{R}^n$. We define the conditioning number of A restricted to V by

$$\kappa_V = \sup_{\mathbf{x}\in V} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|}.$$

The well-conditioned space is the space W such that κ_W is not too large, whereas the ill-conditioned space I is the one where κ_I is very large.

For matrices deriving from the discretization of compact integral operator we have that W corresponds to the low frequency space and I is the high frequency space.

The idea of the Multigrid method is to combine the positive aspects of both direct and iterative method.

Let us start by describing the Two Grid Method (TGM).

The TGM is an iterative algorithm. Let $\mathbf{x}_k$ be an approximation of the solution of (7.1) at the kth step, apply ν_1 steps of an iterative method to $\mathbf{x}_k$ obtaining

$$\tilde{\mathbf{x}}_k = \text{Pre-Smooth}(A, \mathbf{b}, \mathbf{x}_k, \nu_1).$$

This step is called *pre-smoothing*, since it is done before everything else and, in the context of differential equations, damps the error in the high frequencies, i.e., it smooths the error.

We then compute the residual

$$\mathbf{r}_k = \mathbf{b} - A\tilde{\mathbf{x}}_k,$$

since we are moving to the error equation in order to compute a refinement term for $\tilde{\mathbf{x}}_k$. Let $0 < n_1 < n$, we call $R \in \mathbb{R}^{n_1 \times n}$ the restriction operator. This operator projects a vector from a grid of size n to a grid of size n_1.

Define $P \in \mathbb{R}^{n \times n_1}$ the interpolation operator. This operator interpolates a vector from a grid of size n_1 to a grid of size n. Usually $R = P^t$.

We can now define the restricted operator using the Galerkin approach as

$$A_1 = RAP \in \mathbb{R}^{n_1 \times n_1}.$$

Let us assume that both R and P are of full rank. This implies that A_1 is invertible. Computing the refinement term for $\tilde{\mathbf{x}}_k$ as

$$\mathbf{h}_k = PA_1^{-1}R\mathbf{r}_k,$$

we obtain the refined version of $\tilde{\mathbf{x}}_k$ as

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \mathbf{h}_k = \tilde{\mathbf{x}}_k + PA_1^{-1}R(\mathbf{b} - A\tilde{\mathbf{x}}_k).$$

The procedure that computes $\hat{\mathbf{x}}_k$ from $\tilde{\mathbf{x}}_k$ is called Coarse Grid Correction (CGC). Let us call $\mathcal{C}$ the iteration matrix of the CGC, i.e.,

$$\mathcal{C} = I - P(RAP)^{-1}RA,$$

it is possible to show that $\mathcal{C}$ is a projector and hence $\lambda(\mathcal{C}) = \{0, 1\}$, where $\lambda(\mathcal{C})$ denotes the spectrum of $\mathcal{C}$. Therefore, the TGM algorithm, without any smoothing step, can not converge to the solution of (7.1), cf. [24].

Finally, to obtain the $(k+1)$th approximation, we apply ν_2 steps of an iterative method

$$\mathbf{x}_{k+1} = \text{Post-Smooth}(A, \mathbf{b}, \hat{\mathbf{x}}_k, \nu_2),$$

which can be different from the pre-smoother. This is called *post-smoothing*.

It is possible to show that, under mild conditions, this method converges to the solution of (7.1).

The problem of the TGM algorithm is obviously the computation of $\mathbf{h}_k$ since it requires the inversion of A_1 which, if n_1 is large, can be extremely expensive. The multigrid method stems from this observation. Since A_1 can be very large the idea is to restrict consecutively the grid until it is so small that the inversion can be easily performed.

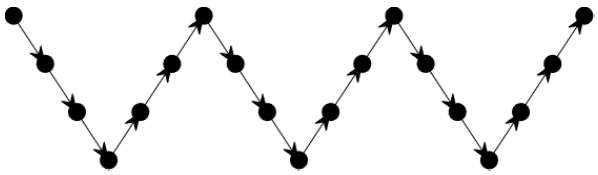


FIGURE 7.1: V-cycle scheme

Let $n = n_0 > n_1 > \cdots > n_L > 0$, we call R_i and P_i the ith restriction and interpolation operator respectively so that $R_i \in \mathbb{R}^{n_{i+1} \times n_i}$ and $P_i \in \mathbb{R}^{n_i \times n_{i+1}}$, for $i = 0, \ldots, L-1$; then

$$A_i = \begin{cases} A & \text{if } i = 0 \\ R_{i-1} A_{i-1} P_{i-1} & i = 1, \ldots, L. \end{cases} \tag{7.2}$$

We proceed to project for L levels then we directly solve the system. The idea is that n_L is so small that A_L can be inverted directly. For instance, for images of size $2^L \times 2^L$ we have $n = 2^{2L}$ and picking up a pixel every two in each direction it holds $n_i = 2^{2(L-i)}$, for $i = 0, \ldots, L$, thus $n_L = 1$.

Summarizing the single step of the multigrid iteration goes as follows

$\mathbf{y}_i = \text{MGM Single Step}(\mathbf{x}_i, \mathbf{b}_i, A_i, i, L)$

```
if (i = L) then y_L   = Solve(A_L y_L = b_L)
           else x̃_i   = Pre-Smooth(A_i, b_i, x_i, ν_1)
                r_{i+1}= R_i(b_i − A_i x̃_i)
                e_{i+1}= MGM Single Step(0, r_{i+1}, A_{i+1}, i + 1, L)
                x̂_i   = x̃_i + P_i e_{i+1}
                y_i   = Post-Smooth(A_i, b_i, x̂_i, ν_2)
end
```

We call this iteration V-cycle since when represented graphically the single iteration resembles a V, see Figure 7.1. Iterating the process we obtain the Multigrid Method

$\mathbf{x} = \text{MGM}(\mathbf{x}_0, \mathbf{b}, A, L)$

```
for k = 1, 2, ...
        x_k = MGM Single Step(x_{k−1}, b, A, 1, L)
end
```

Remark 7.2. *For well-posed problems, in [4] the authors showed that for certain matrix algebras related to BTTB and under some hypothesis that links the smoothers and the coarsening strategy, the Multigrid method has a linear convergence rate, i.e., the number of iterations does not depend on the dimension of the problem.*

Intuitively the idea is that the restriction operator should map into the subspace where A is ill-conditioned. In this way the multigrid is able to deal simultaneously on both the ill-conditioned and the well-conditioned subspaces, on the first the matrix is inverted directly on the second the smoother damps the error very fast.

Conversely, for ill-posed problems, the projection in the ill-conditioned subspace results to be dangerous and the grid transfer operator has to be chosen differently [56]. The reason of this is twofold: firstly the matrix A is usually not invertible and thus, if projected on the ill-conditioned subspace, the restricted operator is not invertible as well; secondly the order

of the zero of the singular values can be very high, even exponential, and thus an eventual (pseudo-)inversion on the ill-conditioned subspace may lead to a dramatical amplification of the noise.

On the other hand the projection in the well-conditioned subspace allows the direct inversion of the coefficient matrix, since, if L is chosen big enough, the restricted operator becomes invertible. Because we assume L large, A_L is very small, usually just a scalar, and so numerical stability is not an issue.

Using a projection into the well-conditioned subspace, the multigrid which we are going to construct does not have a linear convergence rate since it does not fulfills the hypothesis in [4], but it shows a very stable convergence which is also fast thanks to the preconditioned smoother.

7.1.2 Tight frames denoising

We now describe an algorithm for denoising a signal based on the framelet decomposition.
Definition 7.3. *Let $\mathcal{A} \in \mathbb{R}^{r\times n}$ with $n \leq r$, the set of the rows of $\mathcal{A}$ is a* tight frame *for $\mathbb{R}^n$ if $\forall \mathbf{x} \in \mathbb{R}^n$ it holds*

$$\|\mathbf{x}\|^2 = \sum_{\mathbf{y}\in\mathcal{A}} |\langle \mathbf{x}, \mathbf{y}\rangle|^2, \tag{7.3}$$

where $\langle\cdot,\cdot\rangle$ is the inner product of $\mathbb{R}^n$, $\|\cdot\|$ is the Euclidean norm, and $\mathbf{y}$ are the transpose of the rows of $\mathcal{A}$. The matrix $\mathcal{A}$ is the analysis operator *and $\mathcal{A}^*$ is the* synthesis operator.

The equation (7.3) is equivalent to the perfect reconstruction formula

$$\mathbf{x} = \sum_{\mathbf{y}\in\mathcal{A}} |\langle \mathbf{x}, \mathbf{y}\rangle| \mathbf{y} = \mathcal{A}^*\mathcal{A}\mathbf{x}.$$

In other words

$$\mathcal{A} \text{ is a tight frame} \Leftrightarrow \mathcal{A}^*\mathcal{A} = I.$$

Note that in general $\mathcal{A}\mathcal{A}^* \neq I$, unless the system is orthogonal.

Tight frames have been used in many image applications like inpainting and deblurring [31–33]. A very important feature of tight frames is their redundancy. Since the system is redundant the loss of some information can be tolerated.

Moreover, we can identify some of the elements of the tight frame as low frequency vectors and the others as high frequency vectors. In other words we can write

$$\mathcal{A} = \begin{pmatrix} H_0 \\ H_1 \end{pmatrix},$$

where the rows of H_0 are the low frequency vectors and the rows of H_1 are the high frequency vectors. When we apply $\mathcal{A}$ to a vector $\mathbf{x}$ we have

$$\mathcal{A}\mathbf{x} = \begin{pmatrix} H_0 \\ H_1 \end{pmatrix} \mathbf{x} = \begin{pmatrix} \mathbf{c}_0 \\ \mathbf{d}_0 \end{pmatrix}.$$

We can recursively apply this decomposition by decomposing again $\mathbf{c}_0$

$$\mathcal{A}^{(1)}\mathbf{c}_0 = \begin{pmatrix} H_0^{(1)} \\ H_1^{(1)} \end{pmatrix} \mathbf{x} = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{d}_1 \end{pmatrix},$$

where we indicated with $\cdot^{(1)}$ the various operators on the (possibly) smaller space in which $\mathbf{c}_0$ lives. In general we have

$$\begin{cases} \mathbf{c}_0 = \mathbf{x} \\ \mathbf{c}_j = H_0^{(j)} \mathbf{c}_{j-1} & j = 1, \dots, l \\ \mathbf{d}_j = H_1^{(j)} \mathbf{c}_{j-1} & j = 1, \dots, l \end{cases}$$

We want to use this decomposition to eliminate the noise from a signal. In particular, since the noise is a highly oscillating function, the largest components of the noise correspond to the high frequencies. We want to eliminate the noise from this subdomain.

We apply the soft-thresholding technique to the high frequency components $\mathbf{d}_j$.

Let θ be the threshold parameter, the soft-thresholding μ_θ applied to the vector $\mathbf{d}$ is defined as

$$\mu_\theta(\mathbf{d}) = \operatorname{sgn}(\mathbf{d})(|\mathbf{d}| - \theta)_+, \tag{7.4}$$

where by $\operatorname{sgn}(x)$ we denote the sign of x and by x_+ the positive part, i.e., $x_+ = \max\{x, 0\}$, here all the operation are computed element-wise. The choice of the parameter θ is of crucial importance. According to [58] we use

$$\theta = c\sqrt{\frac{\log n}{\sqrt{n}}} \tag{7.5}$$

where $c > 0$ is a constant that for Gaussian noise can be chosen as $c = \frac{\delta}{\|\mathbf{b}^\delta\|}$.

The final algorithm for the denoising applied to l levels is then

Algorithm 7.1 (Denoise)**.** *Let $\mathbf{b}^\delta$ denote the noisy signal, θ the thresholding parameter, and l the number of levels to which apply the denoising.*

$$\mathbf{y} = \textit{Denoise}(\mathbf{z}, \theta, lev, l)$$

$$\begin{aligned} &\texttt{if } lev = l \\ &\qquad \mathbf{y} = \mathbf{z} \\ &\texttt{else} \\ &\qquad \begin{pmatrix} \mathbf{c} \\ \mathbf{d} \end{pmatrix} = \mathcal{A}\mathbf{z} \\ &\qquad \mathbf{c}_1 = \textit{Denoise}(\mathbf{z}, \theta, lev + 1, l) \\ &\qquad \mathbf{d}_1 = \mu_\theta(\mathbf{d}) \\ &\qquad \mathbf{y} = \mathcal{A}^* \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{d}_1 \end{pmatrix} \\ &\texttt{end} \end{aligned} \tag{7.6}$$

In the following we will denote the application of the denoising algorithm to a vector $\mathbf{z}$ by

$$S_\theta^l(\mathbf{z}) := \textsf{Denoise}(\mathbf{z}, \theta, 0, l).$$

The system we are interested in is one of linear B-splines. We will use the corresponding low-pass filter as transfer operator for our multigrid method. In principle it is not necessary to use the same operator for denoising and grid transfer, but the numerical tests show that this combination provides better results. This system is formed by one *low-pass filter* H_0 and

two *high-pass filters* H_1 and H_2, the corresponding masks are

$$\mathbf{h}^{(0)} = \frac{1}{2}(1,\,2,\,1)\,,\quad \mathbf{h}^{(1)} = \frac{\sqrt{2}}{4}(1,\,0,\,-1)\,,\quad \mathbf{h}^{(2)} = \frac{1}{4}(-1,\,2,\,-1)\,.$$

We now derive $\mathcal{A}$ from the masks above; imposing the reflexive boundary condition, so that $\mathcal{A}^*\mathcal{A} = I$, we obtain

$$H_0 = \frac{1}{4}\begin{pmatrix} 3 & 1 & 0 & \dots & 0 \\ 1 & 2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 2 & 1 \\ 0 & \dots & 0 & 1 & 3 \end{pmatrix},\; H_1 = \frac{\sqrt{2}}{4}\begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ 0 & \dots & 0 & -1 & 1 \end{pmatrix},$$

and

$$H_2 = \frac{1}{4}\begin{pmatrix} 1 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ 0 & \dots & 0 & 1 & 1 \end{pmatrix}.$$

These operators are for $1D$ signals, we can define the operators for two-dimensional space by using the tensor product

$$H_{ij} = H_i \otimes H_j, \quad i,j = 0,1,2.$$

Thus we obtain the analysis operator

$$\mathcal{A} = \begin{pmatrix} H_{00} \\ H_{01} \\ \vdots \\ H_{22} \end{pmatrix}.$$

This case is slightly different from the simpler one described above, since there are eight high frequency parts. However, the extension is trivial.

7.2 Our multigrid iterative regularization method

In this Section we describe our algorithmic proposal.

7.2.1 Coarsening

The first thing we discuss is the construction of the matrices A_i. The Galerkin approach in (7.2) is not sure to preserve the structure of the coefficient matrix across the levels. Indeed the proposal in [56] requires images of size $(2^\ell - 1) \times (2^\ell - 1)$, $\ell \in \mathbb{N}$, and zero Dirichlet boundary conditions. In particular if A has a structure defined by reflective or antireflective boundary conditions, $A_1 = R_1^t A P_1$ does not have the same structure. Since preserving the structure of the matrix is essential for fast computations, we want to construct the sequence A_i so that the structure is preserved.

We first define the restriction operator R_i^t and the interpolation operator P_i used in our multigrid. R_i is the *full weighting* operator and for P_i is the *linear interpolation* operator. Let $K_d^{(i)}$ be the downsampling operator at level i and $K_u^{(i)}$ the upsampling operator at level i, usually $K_u^{(i)} = \left(K_d^{(i)}\right)^t$, and define

$$M = \frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}. \tag{7.7}$$

Remark 7.4. *Observe that the mask M is the same of the low-pass filter H_{00} described in Subsection 7.1.2.*

Working on 2D problems we store the data in bi-dimensional arrays. Let $\mathbf{x} \in \mathbb{R}^{n_i \times n_i}$, define the restriction operator $R_i : \mathbb{R}^{n_i \times n_i} \to \mathbb{R}^{n_{i+1} \times n_{i+1}}$ as

$$R_i(\mathbf{x}) = K_d^{(i)}(M^t * \mathbf{x}),$$

where $*$ denotes the convolution operator. Note that $K_d^{(i)}$ is defined as

$$K_d^{(i)} = \tilde{K}_d^{(i)} \otimes \tilde{K}_d^{(i)},$$

where $\tilde{K}_d^{(i)}$ is the one-dimensional down-sampling operator which keeps a component every two. $\tilde{K}_d^{(i)}$ can be written as a $n_{i+1} \times n_i$ matrix, i.e., it is a matrix with more columns than rows. If n_i is even

$$\tilde{K}_d^{(i)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{pmatrix},$$

whereas, if n_i, is odd we obtain

$$\tilde{K}_d^{(i)} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{pmatrix}.$$

Therefore we have

$$R_i(\mathbf{x}) = \left(\tilde{K}_d^{(i)} \otimes \tilde{K}_d^{(i)}\right)(M^t * \mathbf{x}) = \tilde{K}_d^{(i)}(M^t * \mathbf{x})\left(\tilde{K}_d^{(i)}\right)^t,$$

where the one-dimensional down-sampling is applied to each row and each column. Similarly, the prolonging operator $P_i : \mathbb{R}^{n_i \times n_i} \to \mathbb{R}^{n_{i-1} \times n_{i-1}}$ is defined as

$$P_i(\mathbf{x}) = M * \left(K_u^{(i)}\mathbf{x}\right),$$

where $K_u^{(i)}$ is defined as

$$K_u^{(i)} = \tilde{K}_u^{(i)} \otimes \tilde{K}_u^{(i)}.$$

$\tilde{K}_u^{(i)}$ is the one-dimensional up-sampling operator which adds a component every two, i.e., $\tilde{K}_u^{(i)} = \left(\tilde{K}_d^{(i)}\right)^t$. Therefore we have

$$P_i(\mathbf{x}) = M * \left(\tilde{K}_u^{(i)} \otimes \tilde{K}_u^{(i)}\mathbf{x}\right) = M * \tilde{K}_u^{(i)}\mathbf{x}\left(\tilde{K}_u^{(i)}\right)^t.$$

We construct A_i by computing the PSF at level i and then imposing the boundary condition (and so the structure) we want to implement.

$$\mathsf{PSF}_i = \begin{cases} \mathsf{PSF} & i = 1 \\ K_d^{(i)}(M^t * \mathsf{PSF}_{i-1} * M)K_u^{(i)} = R_i\mathsf{PSF}_{i-1}P_i & i = 2, \dots, L. \end{cases} \tag{7.8}$$

In this way we are able to construct operators with the same structures for all levels and achieve fast computations. The matrices PSF_i are computed in a setup phase executed before the iterations of the multigrid method, while the computation of the matrices A_i is not necessary. Note that (7.8) implements a Galerkin approach on the stencil of the PSF coefficients instead of the matrices A_i like in (7.2). This is equivalent to define a sequence of continuous operators independent of the boundary conditions that will be applied.

We project down until we reach a level L such that the system is reduced to a single equation in only one variable. In this way the solution of the system at this level is stable and fast.

7.2.2 Smoothing

The next thing we have to specify is the pre-smoother. Our choice is the wavelet framelet denoising described in Section 7.1.2. With this choice of pre-smoother we are able to keep under control the effect of the noise, while preserving the edges. Differently from [45] here we use the framelet denoising as a pre-smoother instead that as a post-smoother. This choice is mainly due to the fact that we want to project the iteration inside the nonnegative cone. The denoising of a nonnegative signal can, in principle, insert negative values and thus, since the post-smoother is the very last operation performed, using it as a post-smoother may result in that the determined approximation has negative values.

The threshold parameter θ for the denoising is chosen as in (7.5) for the first iteration. However, the post-smoother that we are going to use also has a denoising effect, thus we choose to decrease the parameter throughout the iterations.

Since the pre-smoother acts directly on the initial approximation at each level we have that we are going to denoise only the finest level. In fact the initial approximation of the coarser levels is the zero vector and thus its soft-thresholded version is again $\mathbf{0}$, independently from the parameter θ.

We use the following sequence of parameters

$$\theta_k = p^k \frac{\delta}{\|\mathbf{b}^\delta\|}\sqrt{\frac{\log n}{\sqrt{n}}}, \tag{7.9}$$

where $0 < p < 1$ and k denotes the iteration.

For the post-smoother we want to use one iteration of Algorithm 6.1 AIT described in Chapter 6. For the computation of the regularization parameter we need an estimate of the norm of the noise for each levels. To derive this estimation we refer to [100], where the authors

showed that, indexing with 0 the finest level and L the last one, the norm of the noise δ_i can be estimated as

$$\delta_i = \frac{\delta_{i-1}}{4} \quad i = 1, \dots, L-1, \tag{7.10}$$

where $\delta_0 = \delta$.

As we pointed out before, enforcing the nonnegativity of the solution can help in achieving better reconstructions so we want to be sure that our method fulfills this constraint. This can be easily added at each iteration as shown in [34], which is equivalent to use APIT as post-smoother at the finest level.

Note that it does not make sense to do this kind of projection on every level since, a part from the finest level, we are working on the error equation and so it is harmful to impose the nonnegative constraint at the coarser levels.

7.2.3 The algorithm

After having defined the smoothers and the coarsening strategy, the last thing to discuss is the stopping criterion. To determine at which iteration we want to stop our multigrid regularization, we use the discrepancy principle with a parameter similarly to what used for APIT in Chapter 6. Let $\mathbf{x}_k$ be the approximated solution at step k. Then the stopping iteration k^δ is

$$k^\delta = \min_k \left\{ k : \left\| A\mathbf{x}_k - \mathbf{b}^\delta \right\| \leq \frac{1+2\rho}{1-2\rho}\delta \right\}, \tag{7.11}$$

where ρ is defined in (6.1).

Algorithm 7.2 (MgM)**.** *Consider the system* (2.3)*. Choose suitable boundary conditions and let A_i be defined as the blurring matrix with PSF PSF_i defined in* (7.8) *for $i = 0, \dots, L$. Let the noise levels for each level δ_i be defined as in* (7.10)*, the parameter θ_k be chosen as in* (7.9)*, and choose the number of framelet levels l to which apply the denoising. Let $\mathbf{x}_0$ be an initial guess for the solution of* (2.3)

$$\mathbf{x} = \mathit{MGM}(\mathbf{x_0}, \mathbf{b}^\delta, A)$$

$k = 1$
While $\left\|A\mathbf{x}_k - \mathbf{b}^\delta\right\| > \frac{1+2\rho}{1-2\rho}\delta$
$\quad \mathbf{x_k} = \mathit{MGM\ Single\ Step}(\mathbf{x}_{k-1}, \mathbf{b}^\delta, A, 1, l, L)$
$\quad k = k + 1$
end

The single step of the algorithm is defined as

$$\mathbf{y}_i = \mathit{MGM\ Single\ Step}(\mathbf{x}_i, \mathbf{b}_i^{\delta_i}, A_i, i, l, L)$$

if $(i = L)$ **then** $\mathbf{y}_i = \mathit{Solve}\ A_L\mathbf{y}_L = \mathbf{b}_L^{\delta_L}$
else $\tilde{\mathbf{x}}_i = \begin{cases} S^l_{\theta_k}(\mathbf{b}^\delta) & i = 1 \\ \mathbf{x}_i & \textit{otherwise} \end{cases}$
$\mathbf{r}_{i+1} = P_i^t(\mathbf{b}_i^{\delta_i} - A_i\tilde{\mathbf{x}}_i)$
$\mathbf{e}_{i+1} = \mathit{MGM\ Single\ Step}(\mathbf{0}, \mathbf{r}_{i+1}, A_{i+1}, i+1, l, L)$
$\hat{\mathbf{x}}_i = \tilde{\mathbf{x}}_i + P_i\mathbf{e}_{i+1}$
$\hat{\mathbf{y}}_i = \mathit{AIT}(\hat{\mathbf{x}}_i, A_i, \mathbf{b}_i^{\delta_i}, 1)$
$\mathbf{y}_i = \begin{cases} P_\Omega(\hat{\mathbf{y}}_i) & i = 1 \\ \hat{\mathbf{y}}_i & \textit{otherwise} \end{cases}$
end

Where by $\mathsf{AIT}(\hat{\mathbf{x}}_i, A_i, \mathbf{b}_i^{\delta_i}, 1)$ *we mean that we apply one step of Algorithm 6.1 with initial guess* $\hat{\mathbf{x}}_i$*, system matrix* A_i*, right-hand side* $\mathbf{b}_i^{\delta_i}$*, and we estimate the noise level with* δ_i*.*

Remark 7.5. *As stated in Remark 7.2, if the smother and the projector are chosen in the right way, the multigrid algorithm is optimal. However, our choice does not fulfill the hypothesis in [4]. In particular, the chosen projector does not project into the ill-conditioned space of* A*. On the contrary, being a low-pass filter, project into the well-conditioned space of* A *that is the low-frequency space.*

This means that our algorithm is not optimal and, as we will see in Section 7.4, the number of iterations required is usually slightly higher when compared to the post-smoother APIT. However, this is needed in order to obtain a regularizing effect as we are going to see in Section 7.3.

Concerning the arithmetic cost of one multigrid iteration, this is not much higher than the cost of a single iteration of the post-smoother APIT at the finer level, which is lower than $cn \log n$ for a fixed constant c, up to lower order terms, due to four FFTs (two for computing the residual with the chosen boundary conditions and two for applying the preconditioner). Indeed, recalling that the cost of the denoising pre-smoother at the finest level is linear in n, the computational cost at each level i is lower than $cn_i \log n_i$, for $i = 0 \dots, L$, up to lower order terms. Therefore, the total arithmetic cost of one iteration of our MgM for image deblurring is

$$\frac{4}{3} cn \log n + O(n)$$

according to the computational cost of classical V-cycle [120].

7.3 Convergence Analysis

We are now going to study the convergence and regularization properties of our algorithm. In order to do that, however, we are going to restrict ourselves to the simpler case of the two grid method, i.e., $L = 2$, as is usually done for the theoretical analysis of multigrid methods (see e.g. [103, 120]).

Assume that $P_i(x) = R_i(x)^t$ as in our numerical results and denoting the interpolation operator by P such that

$$A_1 = P^t A P.$$

In this simplified version, the algorithm becomes

Algorithm 7.3 (TGM)**.** *Consider the system* (2.3)*. Let the parameter* θ_k *be chosen as in* (7.9)*, and choose the number of framelet levels* l *to which apply the denoising. Let* ρ *be the parameter in equation* (6.1) *and* q *be a fixed constant such that* $2\rho \leq q \leq 1$*. Let* $\mathbf{x}_0$ *be an initial guess for the solution of*

(2.3)

$\mathbf{x} = \mathit{TGM}(\mathbf{x}_0, \mathbf{b}^\delta, A)$

$k = 0$
`While` $\left\|\mathbf{b}^\delta - A\mathbf{x}_k\right\| \geq \frac{2+\rho}{2-\rho}\delta$
$\tilde{\mathbf{x}}_k = S^l_{\theta_k}(\mathbf{x}_k)$
$\mathbf{r}_k = \mathbf{b}^\delta - A\tilde{\mathbf{x}}_k$
$\mathbf{h}_k = P(P^TAP)^{-1}P^t\mathbf{r}_k$
$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \mathbf{h}_k$
$q_k = \max\left\{q, 2\rho + \frac{(1+\rho)\delta}{\left\|\mathbf{b}^\delta - A\tilde{\mathbf{x}}_k\right\|}\right\}$
$\alpha_k =$ `compute` α_k `such that`
$\left\|\mathbf{r}_k - CC^*(CC^* + \alpha_k I)^{-1}\mathbf{r}_k\right\| = q_k \left\|\mathbf{r}_k\right\|$
$\mathbf{x}_{k+1} = P_\Omega\left(\hat{\mathbf{x}}_k + C^t(CC^t + \alpha_k I)^{-1}(\mathbf{b}^\delta - A\hat{\mathbf{x}}_k)\right)$
$k = k + 1$
`end`

We define the following errors

$$\begin{cases} \mathbf{e}_k = \mathbf{x}^\dagger - \mathbf{x}_k \\ \tilde{\mathbf{e}}_k = \mathbf{x}^\dagger - \tilde{\mathbf{x}}_k \\ \hat{\mathbf{e}}_k = \mathbf{x}^\dagger - \hat{\mathbf{x}}_k \end{cases} \tag{7.12}$$

For convenience we also define

$$D = P(P^tAP)^{-1}P^t \tag{7.13}$$

$$Q_k = C^t(CC^t + \alpha_k I)^{-1}. \tag{7.14}$$

In order to prove the convergence we need the following
Assumption 7.1. *We assume that*

(i) The thresholding parameters θ_k and the number of levels l to which apply the denoising are chosen such that

$$\|\mathbf{e}_k\| \geq \|\tilde{\mathbf{e}}_k\|.$$

(ii) The matrix P^tAP is invertible and the noise does not have any component in $\mathcal{R}(P^t)$, i.e., $P^t\mathbf{b}^\delta = P^t\mathbf{b}$.

Before proving the convergence of the TGM algorithm let us discuss Assumption 7.1. Point (i) asks that the threshold parameter is chosen so that it does not deteriorate the error. While this seems a strong request it is satisfied if the parameter is small enough. Since we have chosen a strictly decreasing sequence it is reasonable to think that the assumption will be satisfied, at least, for k large enough. (ii) is unlikely to be satisfied in a real case scenario, however, our MgM Algorithm 7.2 directly inverts the problem when A_L is a scalar and thus in this case the assumption becomes reasonable as well. In fact, the PSF and M defined in (7.7) are nonnegative, hence A_L can be equal to zero only if the initial PSF is zero.
Remark 7.6. *Note that when the data are exact, i.e., $\delta = 0$, Assumption 7.1(ii) is trivially satisfied. Moreover, we set $\theta_k \equiv 0$ and so also Assumption 7.1(i) holds. In other words in the noise-free case Assumption 7.1 is satisfied.*

First of all we have to prove the following
Lemma 7.7. *Let $\hat{\mathbf{e}}_k$ and $\tilde{\mathbf{e}}_k$ be defined in* (7.12)*. Under Assumption 7.1(ii) it holds*

$$\|\hat{\mathbf{e}}_k\| \leq \|\tilde{\mathbf{e}}_k\|.$$

Proof. We denote by $\mathcal{C} = I - DA$ the Coarse Grid correction matrix, $\mathcal{C}$ is a projector and so has spectral norm equal to 1. Thus, it holds

$$\begin{aligned}
\|\hat{\mathbf{e}}_k\| &= \left\|\mathbf{x}^\dagger - \hat{\mathbf{x}}_k\right\| \\
&= \left\|\mathbf{x}^\dagger - (\tilde{\mathbf{x}}_k + D(\mathbf{b}^\delta - A\tilde{\mathbf{x}}))\right\| \\
&\overset{(a)}{=} \left\|\mathbf{x}^\dagger - (\tilde{\mathbf{x}}_k + D(\mathbf{b} - A\tilde{\mathbf{x}})\right\| \\
&= \left\|\mathbf{x}^\dagger - (\tilde{\mathbf{x}}_k + D(A\mathbf{x}^\dagger - A\tilde{\mathbf{x}}))\right\| \\
&= \left\|\mathbf{x}^\dagger - \tilde{\mathbf{x}}_k - DA(\mathbf{x}^\dagger - \tilde{\mathbf{x}})\right\| \\
&= \left\|(I - DA)(\mathbf{x}^\dagger - \tilde{\mathbf{x}}_k)\right\| \\
&\leq \|\mathcal{C}\| \, \|\tilde{\mathbf{e}}_k\| \\
&= \|\tilde{\mathbf{e}}_k\| \,,
\end{aligned}$$

where (a) is justified by Assumption 7.1(ii); in fact

$$D\mathbf{b}^\delta = P(P^tAP)^{-1}P^t\mathbf{b}^\delta = P(P^tAP)^{-1}P^t\mathbf{b} = D\mathbf{b}.$$

□

We are now in a position to prove

Proposition 7.8. *Let $\hat{\mathbf{e}}_k$, $\tilde{\mathbf{e}}_k$, and $\mathbf{e}_k$ be defined in* (7.12)*. Assume that $\mathbf{x}^\dagger \in \Omega$, under Assumptions 6.1 and 7.1 it holds*

$$\|\mathbf{e}_k\|^2 - \|\mathbf{e}_{k+1}\|^2 \geq 2\rho \left\|(CC^t + \alpha_k I)^{-1}\hat{\mathbf{r}}_k\right\| \|\hat{\mathbf{r}}_k\| \,, \tag{7.15}$$

where $\hat{\mathbf{r}}_k = \mathbf{b}^\delta - A\hat{\mathbf{x}}_k$.

Proof. From Assumption 7.1(i) and Lemma 7.7 we have

$$\|\mathbf{e}_k\|^2 - \|\mathbf{e}_{k+1}\|^2 \geq \|\tilde{\mathbf{e}}_k\|^2 - \|\mathbf{e}_{k+1}\|^2 \geq \|\hat{\mathbf{e}}_k\|^2 - \|\mathbf{e}_{k+1}\|^2 \,.$$

The proof now continues as the proof of Proposition 6.21 in Section 6.3.

Denote with $\hat{\mathbf{h}}_k = C^t(CC^t + \alpha_k I)^{-1}\hat{\mathbf{r}}_k$. Consider

$$\begin{aligned}
\|\mathbf{e}_{k+1}\|^2 &= \left\|\mathbf{x}^\dagger - \mathbf{x}_{k+1}\right\|^2 \\
&= \left\|\mathbf{x}^\dagger - P_\Omega\left(\hat{\mathbf{x}}_k + \hat{\mathbf{h}}_k\right)\right\|^2 \\
&\overset{(a)}{=} \left\|P_\Omega\left(\mathbf{x}^\dagger\right) - P_\Omega\left(\hat{\mathbf{x}}_k + \hat{\mathbf{h}}_k\right)\right\|^2 \\
&\overset{(b)}{\leq} \left\|\mathbf{x}^\dagger - \hat{\mathbf{x}}_k - \hat{\mathbf{h}}_k\right\|^2 \\
&= \|\hat{\mathbf{e}}_k\|^2 - 2\left\langle \hat{\mathbf{e}}_k, \hat{\mathbf{h}}_k \right\rangle + \left\|\hat{\mathbf{h}}_k\right\|^2 ,
\end{aligned}$$

where to obtain (a) we have used the fact that, by assumption, $\mathbf{x}^\dagger \in \Omega$ and for (b) we have used the fact that the metric projection is a contractive mapping (see [121]).

Thus

$$\begin{aligned}
\|\mathbf{e}_k\|^2 - \|\mathbf{e}_{k+1}\|^2 &\geq \|\hat{\mathbf{e}}_k\|^2 - \|\mathbf{e}_{k+1}\|^2 \\
&= 2\left\langle \hat{\mathbf{e}}_k, \hat{\mathbf{h}}_k \right\rangle - \left\| \hat{\mathbf{h}}_k \right\|^2 \\
&= 2\left\langle C\hat{\mathbf{e}}_k, (CC^t + \alpha_k I)^{-1}\hat{\mathbf{r}}_k \right\rangle - \left\langle \hat{\mathbf{r}}_k, CC^t(CC^t + \alpha_k I)^{-2}\hat{\mathbf{r}}_k \right\rangle \\
&= 2\left\langle \hat{\mathbf{r}}_k, (CC^t + \alpha_k I)^{-1}\hat{\mathbf{r}}_k \right\rangle - \left\langle \hat{\mathbf{r}}_k, CC^t(CC^t + \alpha_k I)^{-2}\hat{\mathbf{r}}_k \right\rangle \\
&\quad - 2\left\langle \hat{\mathbf{r}}_k - C\hat{\mathbf{e}}_k, (CC^t + \alpha_k I)^{-1}\hat{\mathbf{r}}_k \right\rangle \\
&\geq 2\left\langle \hat{\mathbf{r}}_k, (CC^t + \alpha_k I)^{-1}\hat{\mathbf{r}}_k \right\rangle - 2\left\langle \hat{\mathbf{r}}_k, CC^t(CC^t + \alpha_k I)^{-2}\hat{\mathbf{r}}_k \right\rangle \\
&\quad - 2\left\langle \hat{\mathbf{r}}_k - C\hat{\mathbf{e}}_k, (CC^t + \alpha_k I)^{-1}\hat{\mathbf{r}}_k \right\rangle \\
&= 2\alpha_k \left\langle \hat{\mathbf{r}}_k, (CC^t + \alpha_k I)^{-2}\hat{\mathbf{r}}_k \right\rangle - \left\langle \hat{\mathbf{r}}_k - C\hat{\mathbf{e}}_k, (CC^t + \alpha_k I)^{-1}\hat{\mathbf{r}}_k \right\rangle \\
&\geq 2\alpha_k \left\langle \hat{\mathbf{r}}_k, (CC^t + \alpha_k I)^{-2}\hat{\mathbf{r}}_k \right\rangle - \|\hat{\mathbf{r}}_k - C\hat{\mathbf{e}}_k\| \left\| (CC^t + \alpha_k I)^{-1}\hat{\mathbf{r}}_k \right\| \\
&= 2\left\| (CC^t + \alpha_k I)^{-1}\hat{\mathbf{r}}_k \right\| \left(\left\| \alpha_k (CC^t + \alpha_k I)^{-1}\hat{\mathbf{r}}_k \right\| - \|\hat{\mathbf{r}}_k - C\hat{\mathbf{e}}_k\| \right)
\end{aligned}$$

Since $\alpha_k(CC^t + \alpha_k I)^{-1}\hat{\mathbf{r}}_k = \hat{\mathbf{r}}_k - C\hat{\mathbf{h}}_k$ and inserting the definition of α_k we have that

$$\left\| \alpha_k (CC^t + \alpha_k I)^{-1}\hat{\mathbf{r}}_k \right\| = q_k \|\hat{\mathbf{r}}_k\| .$$

Thus

$$\|\mathbf{e}_k\|^2 - \|\mathbf{e}_{k+1}\|^2 \geq 2\left\| (CC^t + \alpha_k I)^{-1}\hat{\mathbf{r}}_k \right\| \left(q_k \|\hat{\mathbf{r}}_k\| - \|\hat{\mathbf{r}}_k - C\hat{\mathbf{e}}_k\| \right) .$$

Using Lemma 6.1 and the definition of q_k

$$\begin{aligned}
\|\mathbf{e}_k\|^2 - \|\mathbf{e}_{k+1}\|^2 &\geq 2\left\| (CC^t + \alpha_k I)^{-1}\hat{\mathbf{r}}_k \right\| \left(q_k \|\hat{\mathbf{r}}_k\| - \left(\rho + \frac{1+\rho}{\tau_k} \right) \|\hat{\mathbf{r}}_k\| \right) \\
&\geq 2\rho \left\| (CC^t + \alpha_k I)^{-1}\hat{\mathbf{r}}_k \right\| \|\hat{\mathbf{r}}_k\| .
\end{aligned}$$

which concludes the proof. □

Corollary 7.9. *With the same notation and assumptions as above it holds*

$$\|\mathbf{e}_0\| \geq 2\rho \sum_{k=0}^{k^\delta - 1} \left\| (CC^t + \alpha_k I)^{-1}\hat{\mathbf{r}}_k \right\| \|\hat{\mathbf{r}}_k\| \geq c \sum_{k=0}^{k^\delta - 1} \|\mathbf{r}_{k+1}\|^2 ,$$

for some constant $c > 0$ depending only on ρ and q.

Proof. Corollary 6.3 implies that

$$\|\mathbf{e}_0\| \geq 2\rho \sum_{k=0}^{k^\delta - 1} \left\| (CC^t + \alpha_k I)^{-1}\hat{\mathbf{r}}_k \right\| \|\hat{\mathbf{r}}_k\| \geq \bar{c} \sum_{k=0}^{k^\delta - 1} \|\hat{\mathbf{r}}_k\|^2 .$$

If we prove that $\|\hat{\mathbf{r}}_k\| > d \|\mathbf{r}_{k+1}\|$, where d is a constant depending only on ρ and q, the thesis will follow with $c = d\bar{c}$.

Before proving the main result we give an estimate of $\|AQ_k\|$ for Q_k defined in (7.14). It holds

$$\begin{aligned}\|AQ_k\| &= \sup_{\|\mathbf{z}\|=1} \left\|AC^t(CC^t+\alpha_k I)^{-1}\mathbf{z}\right\| \\ &\leq \frac{1}{1-\rho} \sup_{\|\mathbf{z}\|=1} \left\|CC^t(CC^t+\alpha_k I)^{-1}\mathbf{z}\right\| \\ &\leq \frac{1}{1-\rho},\end{aligned}$$

where we have used the fact that, because of Assumption 6.1 $\forall \mathbf{z}\ \|A\mathbf{z}\| \leq \frac{1}{1-\rho}\|C\mathbf{z}\|$ and that $CC^t(CC^t+\alpha_k I)^{-1}$ is symmetric and has all the eigenvalues between 0 and 1 for any $\alpha_k > 0$.

Finally

$$\begin{aligned}\|\mathbf{r}_{k+1}\| &= \left\|\mathbf{b}^\delta - A\mathbf{x}_{k+1}\right\| \\ &= \left\|\mathbf{b}^\delta - A(\hat{\mathbf{x}}_k + C^t(CC^t+\alpha_k I)^{-1}(\mathbf{b}^\delta - A\hat{\mathbf{x}}_k))\right\| \\ &= \left\|(I - AC^t(CC^t+\alpha_k I)^{-1})\hat{\mathbf{r}}_k\right\| \\ &\leq (1 + \left\|AC^t(CC^t+\alpha_k I)^{-1}\right\|)\,\|\hat{\mathbf{r}}_k\| \\ &\leq \left(2 + \frac{1}{1-\rho}\right)\|\hat{\mathbf{r}}_k\|,\end{aligned}$$

which concludes the proof. □

Corollary 7.9 shows that, when $\delta > 0$, Algorithm 7.3 stops after finitely many iterations, independently of the choice of $\mathbf{x}_0$. In fact, assume that $k^\delta = \infty$, i.e. that the algorithm does not stop after finitely many iterations. Then we would have that

$$\sum_{k=0}^{\infty} \|\mathbf{r}_k\|^2 \leq \|\mathbf{e}_0\| < \infty.$$

Thus the norm of the residual becomes arbitrarily small and in particular smaller than $\frac{1+2\rho}{1-2\rho}\delta$ which is absurd.

We are now in a position to prove the convergence of Algorithm 7.3 in the noise free case and that, if $\mathbf{x}_0$ is not a solution of the system, an infinite number of iterations are needed.

Theorem 7.10. *Let $\delta = 0$ and suppose that $\mathbf{x}_0$ is not a solution of the system* (2.3)*. Then the iterates generated by Algorithm 7.2 converge to a solution of* (2.3)*. Moreover, an infinite number of iterations are needed.*

Proof. This proof is in the spirit of the proof of [49, Theorem 4], however some details are different and thus we show it here.

We start first by proving that infinitely many iterations are needed. If $\delta = 0$ then the stopping criterion can be satisfied for a $k = k^\delta$ such that $\mathbf{x}_k$ is a solution of the system (2.3). This means that calling

$$\hat{\mathbf{h}}_{k-1} = C^t(CC^t+\alpha_k I)^{-1}\hat{\mathbf{r}}_{k-1},$$

it should coincide with $\hat{\mathbf{e}}_{k-1}$ up to an element in the null space of A, which is the null space of C because of Assumption 6.1. From the definition of α_{k-1} and Lemma 6.1 it follows that

$$q_{k-1}\|\hat{\mathbf{r}}_{k-1}\| = \left\|\hat{\mathbf{r}}_{k-1} - C\hat{\mathbf{h}}_{k-1}\right\| = \|\hat{\mathbf{r}}_{k-1} - C\hat{\mathbf{e}}_{k-1}\| \leq \left(\rho + \frac{1+\rho}{\tau_{k-1}}\right)\|\hat{\mathbf{r}}_{k-1}\|.$$

However, this contradicts the definition of q_{k-1} which means that the iterations does not stop if $\hat{\mathbf{x}}_0$ is not a solution of the system.

We now show that the sequence $\{\mathbf{x}_k\}_k$ converges.

We first show that the norm of the iterates $\mathbf{x}_k$ is bounded. Observe that

$$\|\mathbf{e}_k\|^2 = \left\|\mathbf{x}_k - \mathbf{x}^\dagger\right\|^2 \geq \|\mathbf{x}_k\|^2 - \left\|\mathbf{x}^\dagger\right\|^2.$$

Using the fact that $\|\mathbf{e}_{k+1}\|^2 \leq \|\mathbf{e}_{k+1}\|^2$ for all k we have that

$$\|\mathbf{e}_0\|^2 \geq \|\mathbf{x}_k\|^2 - \left\|\mathbf{x}^\dagger\right\|^2,$$

this yields

$$\|\mathbf{x}_k\|^2 \leq \left\|\mathbf{x}^\dagger\right\|^2 + \|\mathbf{e}_0\|^2. \tag{7.16}$$

We now need to relate the norm of $\hat{\mathbf{h}}_k$ with $\|\mathbf{r}_k\|^2$. First of all note that, by construction, it holds that $\hat{\mathbf{h}}_k \in \mathcal{R}\left(C^t\right) = \mathcal{N}\left(C\right)^\perp$, thus

$$\hat{\mathbf{h}}_k = C^\dagger C \hat{\mathbf{h}}_k.$$

The equality above yields

$$\begin{aligned}
\left\|\hat{\mathbf{h}}_k\right\| &= \left\|C^\dagger C \hat{\mathbf{h}}_k\right\| \\
&\leq \left\|C^\dagger\right\| \left\|C\hat{\mathbf{h}}_k\right\| \\
&= \left\|C^\dagger\right\| \left\|CC^t\left(CC^t + \alpha_k I\right)^{-1}\hat{\mathbf{r}}_k\right\| \\
&\overset{(a)}{\leq} \left\|C^\dagger\right\| \|\hat{\mathbf{r}}_k\| \\
&\overset{(b)}{\leq} c\left\|C^\dagger\right\| \|\mathbf{r}_{k+1}\|,
\end{aligned} \tag{7.17}$$

where inequality (a) is obtained by observing that $CC^t\left(CC^t + \alpha_k I\right)^{-1}$ is a symmetric matrix whose eigenvalues are all smaller than 1 for any $\alpha_k > 0$ and (b) has been shown in the proof of Corollary 7.9 for some fixed constant $c \geq 0$.

We now show that $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \to 0$ as $k \to \infty$. Consider

$$\begin{aligned}
\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 &= \left\|P_\Omega\left(\mathbf{x}_k + P^t\left(P^tAP\right)^{-1}P\mathbf{r}_k + \hat{\mathbf{h}}_k\right) - \mathbf{x}_k\right\|^2 \\
&\leq \left\|\mathbf{x}_k + P^t(P^tAP)^{-1}P\mathbf{r}_k + \hat{\mathbf{h}}_k - \mathbf{x}_k\right\|^2 \\
&= \left\|P^t(P^tAP)^{-1}P\mathbf{r}_k + \hat{\mathbf{h}}_k\right\|^2 \\
&\leq 2\left\|P^t(P^tAP)^{-1}P\right\|^2 \|\mathbf{r}_k\|^2 + 2\left\|\hat{\mathbf{h}}_k\right\|^2 \\
&\leq 2\left\|P^t(P^tAP)^{-1}P\right\|^2 \|\mathbf{r}_k\|^2 + 2c\left\|C^\dagger\right\|^2 \|\mathbf{r}_{k+1}\|^2,
\end{aligned}$$

where in the last step we have used (7.17). Since $\|\mathbf{r}_k\| \to 0$ as $k \to 0$ in force of Corollary 7.9, we have that

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \to 0 \text{ as } k \to 0. \tag{7.18}$$

Combining (7.16) and (7.18) we have that the $\mathbf{x}_k$ converges to a limit $\mathbf{x}$. Moreover, the residuals $\mathbf{b} - A\mathbf{x}_k \to \mathbf{b} - A\mathbf{x}$, while their norm converges to 0 because of Corollary 7.9. We then have that $A\mathbf{x} = \mathbf{b}$. Thus the limit point of Algorithm 7.3 is a solution of (2.3) □

The last result that we would like to prove is that Algorithm 7.3 is a regularization method.

Theorem 7.11. *Assume that Assumption 6.1 holds for some $0 < \rho \leq \frac{1}{2}$ and let $\delta \mapsto \mathbf{b}^\delta$ be a function from $\mathbb{R}^+$ to $\mathbb{R}^N$ such that for all δ it holds $\|\mathbf{b} - \mathbf{b}^\delta\| \leq \delta$. For fixed τ and q denote by $\mathbf{x}^\delta$ the approximation of $\mathbf{x}^\dagger$ obtained with Algorithm 7.3. Then, as $\delta \to 0$, $\mathbf{x}^\delta$ goes to a solution of the system.*

We omit the proof since it can be copied from [77, Theorem 2.3]; for further reference see also [61, Theorem 11.5]. Its essentials ingredients are the monotonicity proved in Proposition 6.2, the convergence to the exact solution in the exact data case proved in Theorem 7.10 and the continuity of the map $\delta \mapsto \mathbf{b}^\delta$.

7.4 Numerical Examples

We now give some numerical examples.

The only things that are left to define in Algorithm 7.2 are how we construct the approximation C_i of A_i, the parameters ρ_i and q_i. We set C_i, like in Chapter 6, as the blurring matrix with PSF PSF_i, but with periodic boundary conditions. In this way the computation of $C_i^t(C_iC_i^t + \alpha I)^{-1}$ can be done in $O(n \log n)$ flops using the FFT. We set $\rho_i = 10^{-4}$ and $q_i = 0.7$ for all levels.

We compare Algorithm 7.2 to several methods from the literature with respect to both accuracy and efficiency. For the comparison in accuracy we consider the RRE and the Peak Signal to Noise Ratio (PSNR), the latter is defined as follows

$$PSNR(\mathbf{x}) = 20 \log_{10} \left(\frac{\sqrt{n} M}{\|\mathbf{x} - \mathbf{x}^\dagger\|} \right),$$

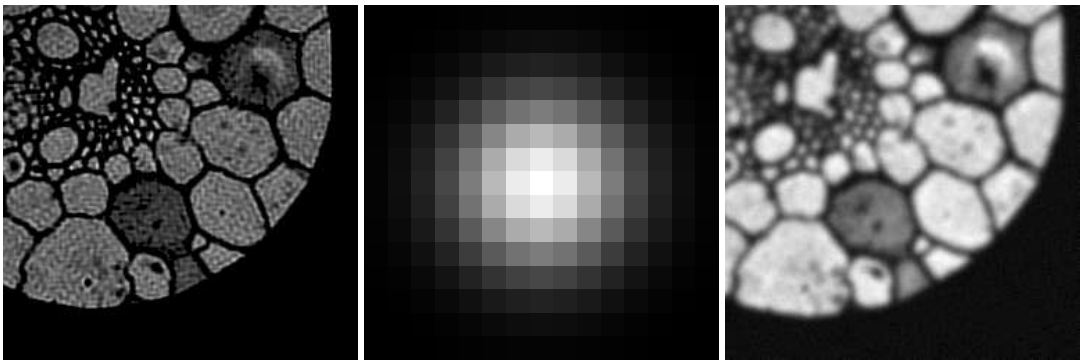
where n is the the number of elements of $\mathbf{x}$ and M denotes the maximum value that can be achieved by $\mathbf{x}^\dagger$.

We compare our MgM Algorithm 7.2 with the following methods

- ADMM with unknown boundary conditions (ADMM-UBC) with Total Variation penalty term, see [2, 3];
- Approximated Projected Iterated Tikhonov (APIT), see Section 6.3;
- Flexible Arnoldi Tikhonov (FlexiAT), see [67];
- Non Negative Restarted Generalized Arnoldi Tikhonov (NN-ReStart-GAT), see [67];
- Range Restricted Arnoldi Tikhonov (RRAT), see [104];
- Two step Iterative Soft Thresholding (TwIST), see [15].

Some of these methods require the estimation of a parameter, in particular this is true for ADMM-UBC, FlexiAT, and TwIST. For these methods we use the parameter which minimizes the RRE (or, equivalently, maximizes the PSNR).

The maximum number of iterations is fixed at 500 for all methods.



(a) (b) (c)

FIGURE 7.2: Grain test problem: (a) True image (242×242 pixels), (b) Gaussian PSF with variance $\sigma = 2.5$ (15×15 pixels), (c) Blurred and noisy image with $\nu = 0.03$ (242×242 pixels).

Method	RRE	PSNR	Iterations
MgM	**0.20479**	**24.9609**	99
ADMM-UBC	0.20982	24.75	315
APIT	0.21582	24.5052	81
FlexiAT	0.25566	23.0338	500
NN-ReStart-GAT	0.26845	22.6096	59
RRAT	0.23263	23.8536	7
TwIST	0.23052	23.933	99

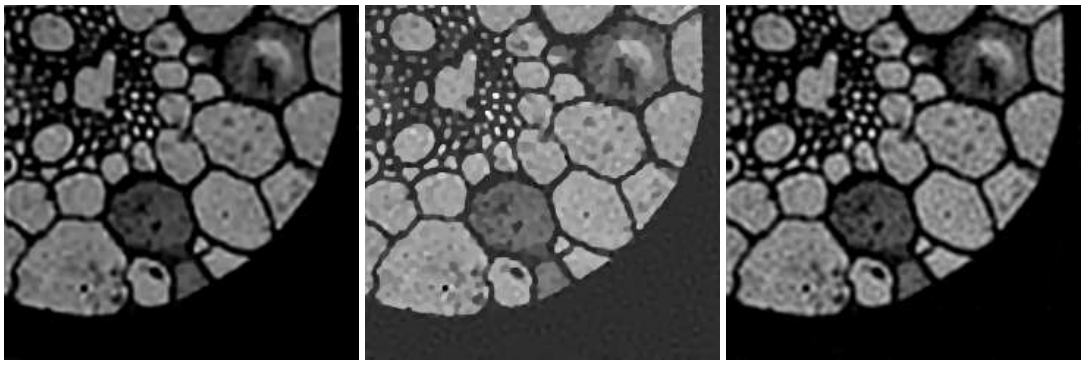
TABLE 7.1: Grain test problem: Comparison between MgM and other methods from the literature. For ADMM-UBC, FlexiAT, and TwIST the optimal regularization parameter was used. In bold the smallest error and the greatest PSNR.

Grain In this first example we consider the grain image and we blur it with a Gaussian PSF with variance $\sigma = 2.5$ and add noise with $\xi = 0.03$. In Figure 7.2 we show the true image, the PSF and the blurred and noisy image. For this example we have employed the *reflexive* boundary conditions.

In Table 7.1 we compare the results obtained with our algorithm against the one obtained with other methods from the literature. MgM gives the best result in term of accuracy while keeping a reasonable computational cost. In Figure 7.3 we can see different reconstruction obtained with three methods: MgM, ADMM-UBC, and APIT. From a visual inspection we can see that the reconstruction obtained with ADMM-UBC is not able to reconstruct the black area in the low right corner. This is due to the fact that ADMM-UBC does not enforce nonnegativity on the reconstruction and thus negative values appears in that area.

We want to stress the fact that MgM does not require the estimation of any parameter whereas ADMM-UBC needs the evaluation of a regularization parameter. In Figure 7.4 we show the variation of the error obtained with ADMM-UBC when the regularization parameter changes. We can see that if the parameter is not estimated accurately the error can become very large.

Cameraman In this second example we blur the cameraman image with a circular blur PSF and add Gaussian noise so that $\xi = 0.02$. In Figure 7.5 we report the true image, the PSF, and the blurred and noisy image. We employ the antireflective boundary conditions. In Table 7.2 we show the results obtained with MgM and the benchmark methods. From this comparison



(a) (b) (c)

FIGURE 7.3: Grain test problem reconstructions obtained with different methods: (a) MgM, (b) ADMM-UBC, (c) APIT.

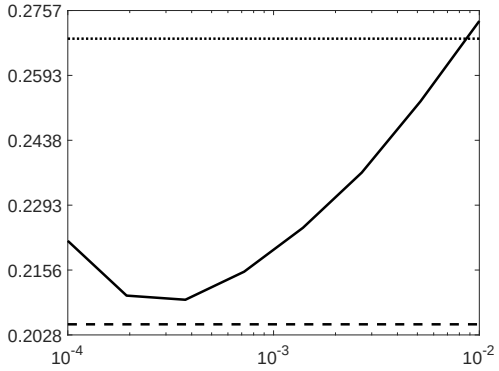
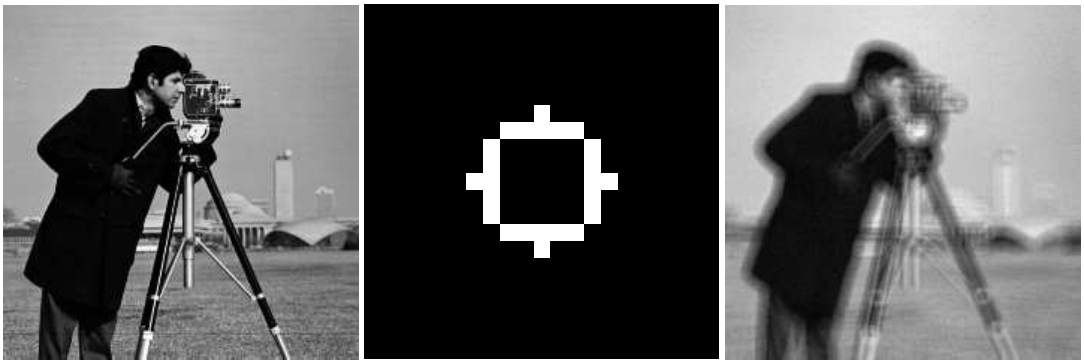


FIGURE 7.4: Grain test problem: Error obtained with ADMM-UBC with respect to the regularization parameter. The solid black line represent the error obtained with ADMM-UBC for different choice of the regularization parameter, the dashed line is the error obtained with MgM, and the dotted line is the RRE obtained with NN-Restart-GAT.



(a) (b) (c)

FIGURE 7.5: Cameraman test problem: (a) True image (238×238 pixels), (b) Circular motion PSF (21×21 pixels), (c) Blurred and noisy image with $\xi = 0.02$ (238×238 pixels).

Method	RRE	PSNR	Iterations
MgM	**0.084219**	**27.1115**	50
ADMM-UBC	0.17452	20.7828	167
APIT	0.11638	24.3018	8
FlexiAT	0.12042	24.0055	9
NN-ReStart-GAT	0.11249	24.5978	58
RRAT	0.11403	24.4795	7
TwIST	0.11311	24.5498	52

TABLE 7.2: Cameraman test problem: Comparison between MgM and other methods from the literature. For ADMM-UBC, FlexiAT, and TwIST the optimal regularization parameter was used. In bold the smallest error and the greatest PSNR.

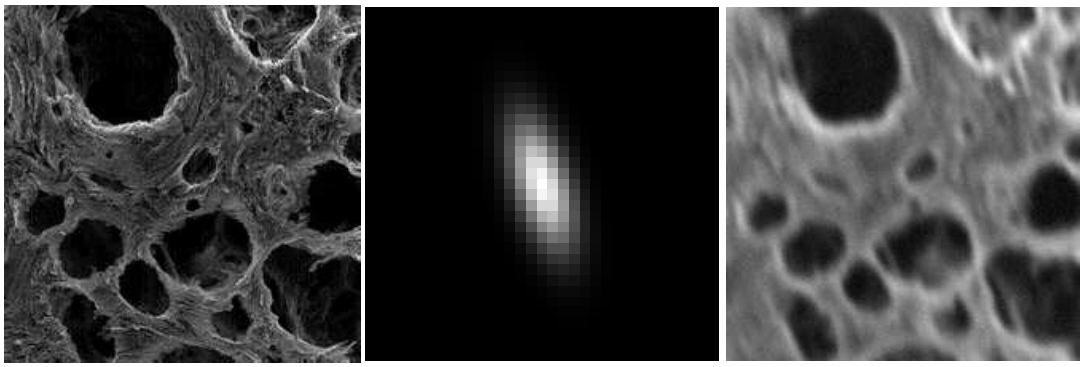
we can see that MgM greatly outperforms all the other methods in term of accuracy, while keeping a reasonable computational time. From the visual inspection of the reconstruction in Figure 7.6 we can see how good the approximation given by MgM also in the small details in the background, while the other methods are affected by a very heavy ringing effect.

Biological Image For the last example we use a biological image and we blur it with a non-symmetric Gaussian PSF, finally we add white Gaussian noise with $\xi = 0.05$. In Figure 7.7 we show the true image, the PSF, and the blurred and noisy image. For the deblurring we use the antireflective boundary conditions.

From the comparison in Table 7.3 of the results obtained with MgM and the other methods considered we can see that our methods outperforms all the others while keeping a reasonable computational cost. Moreover, from the visual inspection of the reconstructions in Figure 7.8 we can see the benefit of the nonnegative constraint. In fact the reconstruction provided by ADMM-UBC looks grayish due to the presence of negative values, whereas the reconstruction obtained using MgM and APIT do not suffer of this problem.

(a) (b) (c)

FIGURE 7.6: Cameraman test problem reconstructions obtained with different methods: (a) MgM, (b) TwIST, (c) NN-Restart-GAT.

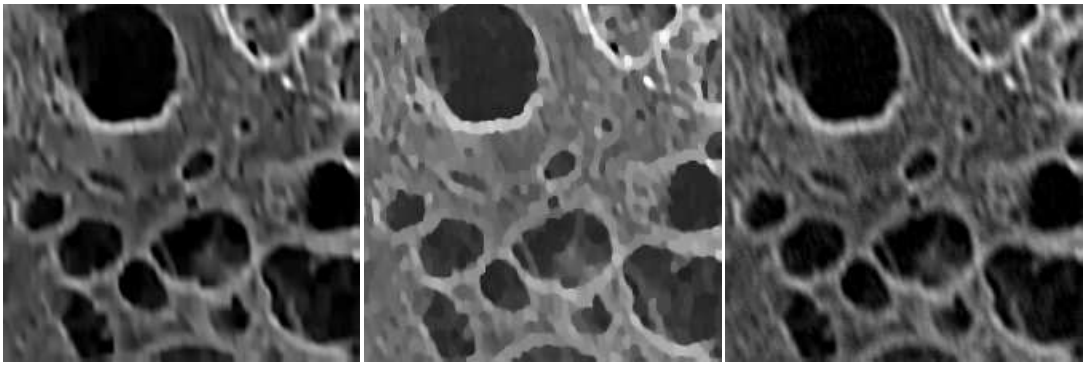


(a) (b) (c)

FIGURE 7.7: Biological image test problem: (a) True image (224×224 pixels), (b) Gaussian non-symmetric PSF (33×33 pixels), (c) Blurred and noisy image with $\xi = 0.05$ (224×224 pixels).

Method	RRE	PSNR	Iterations
MgM	**0.27444**	**22.7457**	47
ADMM-UBC	0.27674	22.6733	333
APIT	0.28328	22.4703	10
FlexiAT	0.31601	21.5208	500
NN-ReStart-GAT	0.35159	20.5939	58
RRAT	0.28603	22.3865	5
TwIST	0.28659	22.3693	53

TABLE 7.3: Biological image test problem: Comparison between MgM and other methods from the literature. For ADMM-UBC, FlexiAT, and TwIST the optimal regularization parameter was used.In bold the smallest error and the greatest PSNR.



(a) (b) (c)

FIGURE 7.8: Biological image test problem reconstructions obtained with different methods: (a) MgM, (b) ADM-UBC, (c) APIT.

Chapter 8

Weakly Constrained Lucy-Richardson algorithm with application to light scattering inversion

In this chapter we are going to consider inverse problems of the form

$$I(\theta) = \int K(\theta, R)\, N(R) \mathrm{d}R, \tag{8.1}$$

coming from optics [19, 70–72, 74], where the main goal of the optical technique is to recover the particle-size distribution $N(R)$ which characterizes the sample under investigation. Thus the solution $N(R)$ is defined to be nonnegative ($N \geq 0$) with a positive support ($R > 0$). Among the various optical techniques, the measurement of the light intensity distribution $I(\theta)$ scattered by the sample at different angles θ, is definitely among the most popular ones [19, 70–72, 74]. Such a technique can be easily implemented by using simple experimental setups [74], with the possibility of characterizing simultaneously a very large number of particles, the characterization being carried out *in situ* and almost *real time* [19, 70–72, 74].

In this setting the goal of an efficient, well performing inversion algorithm is the accurate and fast recovery of the sample particle-size distribution over the largest possible range of particle radii. Indeed, inversion algorithms are expected to work pretty well only when the particle sizes to be recovered lie within a given range $[R_{\min}, R_{\max}]$, which depends on the range $[\theta_{\min}, \theta_{\max}]$ of the independent variable being probed. In the past it has been shown [65] that the dynamical extension of the $R-$range, i.e., the ratio $[R_{\max}/R_{\min}]$, which can be probed, scales proportionally to $[\theta_{\max}/\theta_{\min}]$ and, therefore can be rather limited if such a ratio is not sufficiently large. Typically in optics, $[R_{\max}/R_{\min}] \sim [\theta_{\max}/\theta_{\min}] \sim 10 - 100$, see [65].

Discretizing (8.1) we obtain a linear system

$$A\mathbf{N} = \mathbf{I}, \tag{8.2}$$

where $\mathbf{N} \in \mathbb{R}^m$ and $\mathbf{I} \in \mathbb{R}^n$ are the discretizations of $N(R)$ and $I(\theta)$, respectively, and $A \in \mathbb{R}^{n\times m}$. As mentioned above, since K is compact, the inverse problem in (8.1) is ill-posed.

For solving this problem we are going to consider one of the most popular iterative methods used in this framework: the classical Lucy-Richardson (LR) method [97, 113]. This algorithm has the remarkable feature of ensuring nonnegativity of the solutions. LR is also quite simple, robust against noise and, provided that the iterative procedure is stopped after a large

enough number of steps, does not require any parameter to be optimized. However, as mentioned above and reported in [65][1] when the range of recoverable radii is too large, LR may might be not so efficient.

The method proposed in Chapter 3 looks appealing for the problem describe above, however, it is difficult to apply in the physical setup we consider here. In fact, the norm of the noise, whose knowledge is required by Algorithm 3.4, here is not known and only some statistical informations are available. Moreover, the noise we consider is not white Gaussian and, thus, applying in this framework the standard discrepancy principle is not straightforward.

In this chapter we are going to improve the LR method by adding to the true solution a weak constraint associated to a physical property (the particle volume fraction concentration) which is known or can be measured with high accuracy. The introduction of this additional knowledge largely improves the quality of the restoration and allows to enlarge the $R-$range by more than one order of magnitude, property that is crucial in many real applications. On the other hand, the added weak constraint requires the estimation of a damping parameter which can be easily and robustly optimized by exploiting a second physical property (the particle number concentration), whose value needs to be estimated only very roughly. Note that the strong imposition of the constraint, which is equivalent to a huge value of the damping parameter, does not provide an improvement in the quality of the restoration while requires a larger number of iterations of the convergence, cf. numerical results in Section 8.3. In particular, due to unavoidable measurement errors in the value of the constraint, classical optimization algorithms for linear constrained problems does not provide restorations better than our WCLR, which is simple to implement and does not require any parameter setting in the same spirit of LR.

This chapter is structured as follows. In Section 8.1 we give some physical details about the problem we are going to analyze. In particular, Section 8.1.1 describes the discretization process, while Section 8.1.2 considers the constrains we are going to use. Section 8.2 is devoted to the formulation of the mathematical model and to the definition of our numerical method, which is tested and compared with LR on some numerical examples in Section 8.3.

8.1 Physical details

In this section we give some insight into the physical problem we are going to consider, i.e., the problem associated to the inversion of elastic light scattering (ELS) data, where the main goal is to recover the size distribution of the particles present in the sample.

According to ELS theory [93], when a sample made of polydisperse particles characterized by a refraction index different from that of the surrounding medium is illuminated with a laser light of wavelength λ, part of the radiation is going to be scattered at angles different from the incident direction. If the particles are homogeneously dispersed in the medium and their concentration is so low that they can be considered as non-interacting, the angular distribution of the overall scattered intensity, $I(\theta)$, is given by the sum of the intensities scattered by the single particles [93]. Thus the system is linear and $I(\theta)$ can be written as (8.1) where θ is the scattering angle (the angle between the incident laser beam and the direction at which the scattered light is detected), $N(R)$ is the unknown number-concentration density $[cm^{-3}\,\mu m^{-1}]$ of particles of radius R, and $K(\theta, R)$ is the (known) kernel of the system,

[1] After the publication of [65], it was realized that the method called "modification of the Chahine algorithm" proposed in that work, is identical to the LR algorithm.

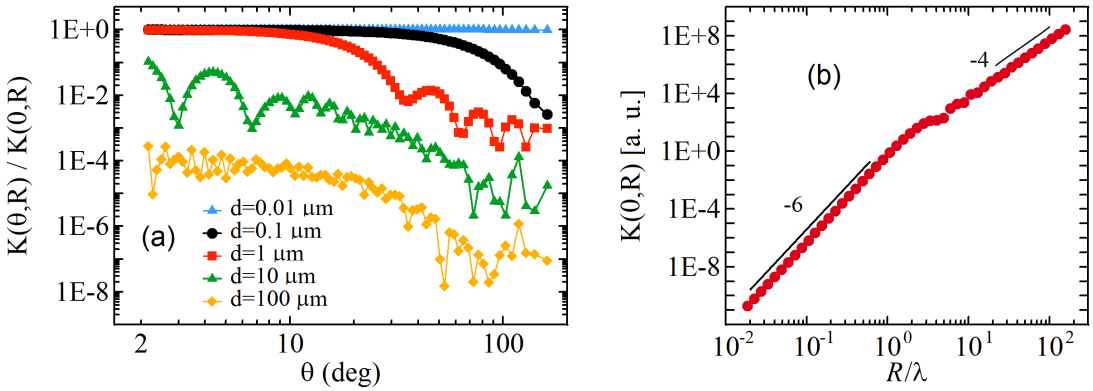


FIGURE 8.1: (a): normalized behavior of the kernel $K(\theta,\, R)$ appearing in (8.1) as a function of the scattering angle θ for five particles with diameters $d = 2R$ ranging from $d = 0.01\,\mu m$ to $d = 100\,\mu m$; the normalization is such that $K(\theta = 0,\, R) = 1$. (b): behavior of the kernel amplitude $K(\theta = 0,\, R)$ as a function of the ratio R/λ. The straight lines with slopes 6 and 4 indicate that $K(\theta = 0,\, R)$ growths as $\sim R^6$ or $\sim R^4$, the crossover occurring at $R \sim \lambda$.

representing the intensity scattered by a single particle of radius R at angle θ. Typically, $I(\theta)$ is detected at a finite number of angles θ_i $(i = 1, .., n)$ within a bounded interval $[\theta_{\min}, \theta_{\max}]$.

If the particles are spheres, the kernel $K(\theta, R)$ is provided by the Mie theory [91], according to which the angular distribution of $I(\theta)$ scattered by a particle of radius R is mostly confined to the diffraction lobe $\theta_{\text{diff}} \sim \lambda/2R$. Moreover, the amplitude of $I(\theta)$ strongly increases with particle radius as $I \sim R^6$ for small particles ($R \ll \lambda$) and $I \sim R^4$ for large particles ($R \gg \lambda$). Figure 8.1(a) reports an example of the behaviors of $I(\theta)/I(0) = K(\theta, R)/K(0, R)$ versus θ over a range of $[2 - 180\,deg]$ for particles of different diameters $d = 2R$ from $d = 0.01$ to $100\,\mu m$. As one can notice, for small particles $I(\theta)$ tends to be rather flat, whereas for large particles $I(\theta)$ exhibits many oscillations and decays by many order of magnitude over the reported θ range. At the same time the zero-angle amplitude $I(0)$ varies widely, passing from $I(0) \sim 10^{-11}$ at $R/\lambda = 10^{-2}$ to $I(0) \sim 10^{8}$ at $R/\lambda = 10^{2}$, see Figure 8.1(b). Thus, it is clear that the inversion of (8.1) might become an unbearable task when the particle size distribution $N(R)$ to be recovered contains particles with very different radii.

8.1.1 Discretization of the Fredholm Integral Equation

We now describe the discretization of (8.1). Let us consider that only a finite number of θ_i $(i = 1, ..., n)$ can be accessed experimentally and within a limited range $[\theta_1, \theta_n]$. Thus, if the particle size distribution $N(R)$ is approximated by a histogram constituted by m bins (or classes) delimited by the radii r_j, $j = 1, ..., m$, the equation (8.1) becomes

$$I(\theta_i) = \sum_{j=1}^{m} A_{ij}\, N_j, \qquad i = 1, 2, \ldots, n, \tag{8.3}$$

where N_j is the number-concentration density $[cm^{-3}\,\mu m^{-1}]$ of the particles belonging to the j-th class of width $\Delta r_j = r_j - r_{j-1}$ and

$$A_{ij} = \int_{r_{j-1}}^{r_j} K(\theta_i, r)\, \mathrm{d}r. \tag{8.4}$$

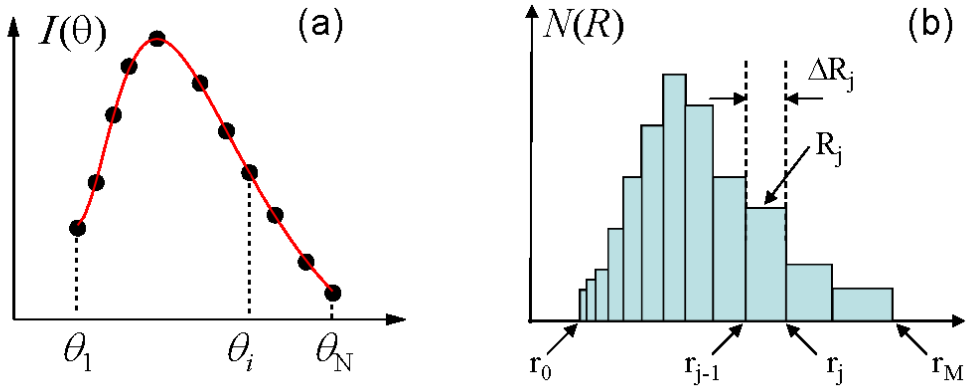


FIGURE 8.2: Discretization scheme of equation (8.1)

When the classes are narrow enough, we can pinpoint them in terms of their average radius $R_j = (r_j + r_{j-1})/2$ and width $\Delta R_j = \Delta r_j = r_j - r_{j-1}$. Thus $N_j \Delta R_j$ represents the number concentration of particles belonging to the $j - th$ class and the term $A_{ij}/\Delta r_j$ is the average intensity scattered at angle θ_i by a single particle with average radius R_j. Note that (8.3) is a set of n linear equations in which the left-hand sides $I(\theta_i)$ are the data provided by the experiment, the matrix entries A_{ij} are known and N_j are the unknowns to be recovered.

Although somewhat arbitrary, it is often convenient to choose the r_j grid so that, within the range $[r_0, r_m]$, all the m classes are characterized by the same relative width $\alpha = \Delta R_j/R_j$. This can be accomplished by scaling r_j according to the geometrical progression

$$r_j = r_0\, a^j,$$

where $a = (R_m/R_1)^{1/(m-1)}$ and $r_0 = 2R_1/(1 + a)$. In this way the average radius and the width of each class scale as

$$R_j = R_1\, a^{j-1} \qquad \Delta R_j = \Delta R_1\, a^{j-1} \qquad j = 1, \ldots, m, \tag{8.5}$$

so that $\alpha = 2[(a - 1)/(a + 1)]$ and for $a \gtrsim 1$, $\alpha \approx a - 1$. A sketch of the classes layout and discretization scheme is reported in Figure 8.2. Typically if we want to cover three order of magnitude in size, i.e., $R_m/R_1 = 10^3$, with $\alpha = 0.02$ then approximately $m = 350$ classes are necessary.

8.1.2 Constraints

The very first constraint we would like to impose is that

$$N_j \geq 0, \qquad j = 1, \ldots, m, \tag{8.6}$$

this comes from the simple observation that the number of particle can not be negative.

A more interesting constraint is related to the integral of $N(R)$

$$c_N = \sum_{j=1}^{m} N_j \Delta R_j, \tag{8.7}$$

where c_N represents the particle number-concentration [cm^{-3}], i.e., the total number of particles contained in the sample divided by the sample volume. This constraint can be applied whenever the number of particle can be counted, a somewhat difficult task that can be carried out only under some experimental conditions.

The last constraint is related to the particle volume fraction concentration c_V, which is given by the total volume occupied by the particles divided by the sample volume

$$c_V = \sum_{j=1}^{m} N_j v_j \Delta R_j, \tag{8.8}$$

where

$$v_j = \frac{1}{\Delta R_j} \int_{R_j - \Delta R_j/2}^{R_j + \Delta R_j/2} (4/3)\pi R^3 \mathrm{d}R \tag{8.9}$$

is the average volume of one particle belonging to the j^{th} class. Clearly, for very narrow classes ($\Delta R_j / R_j \ll 1$), $v_j \approx (4/3)\pi R_j^3$. The c_V constraint is of particular significance because in most experiments the volume concentration is a quantity that can be measured quite easily and with high accuracy.

In our method we would like to exploit both concentration constraints (8.7) and (8.8) (the positiveness constraint, (8.6), being fulfilled automatically, see below), but they are not equivalent from a physical point of view. As mentioned out above, whereas an accurate value of c_V can be easily obtained experimentally, the estimate of c_N might be somewhat troublesome and affected by large errors. Thus, we propose a weakly constrained version of the LR algorithm based on the c_V constraint alone, and we will use the estimate of c_N only for cross-checking the self-consistency of the inversion procedure, i.e., for estimating the damping parameter that weight the constraint on c_V.

8.2 An iterative method based on Lucy-Richardson method

We are now going to describe how the physical model translates into the linear algebra language. The linear system (8.3) is compactly rewritten as

$$A\mathbf{N} = \mathbf{I} \tag{8.10}$$

where $A \in \mathbb{R}^{n \times m}$, $\mathbf{N} \in \mathbb{R}^m$ and $\mathbf{I} \in \mathbb{R}^n$.

Similarly, the three constraints of Section 8.1.2 can be rewritten as:

(i) From (8.6) $\mathbf{N} \geq 0$, meaning that $N_j \geq 0$, for $j = 1, \ldots, m$;

(ii) From (8.7)

$$c_N = \mathbf{N}^t \Delta\mathbf{R}, \tag{8.11}$$

where $0 < \Delta\mathbf{R} \in \mathbb{R}^m$, with $(\Delta\mathbf{R})_j = \Delta R_j$ defined in (8.5) for $j = 1, \ldots, m$;

(iii) From (8.8)

$$c_V = \mathbf{N}^t \mathbf{V} = \sum_{j=1}^{m} V_j N_j. \tag{8.12}$$

where $0 < \mathbf{V} \in \mathbb{R}^m$ and $V_j = v_j \Delta R_j$, being v_j is defined in (8.9).

This information will be used in the following to define a simple and effective iterative procedure to compute a solution of (8.10), where, except from the positiveness (i), the constraints (ii) and (iii) are not all strictly satisfied, but are used to improve the computed approximation or to estimate possible parameters.

Note that equations (8.11) and (8.12) can be seen as weighted ℓ_1-norms because both $\mathbf{V}$ and $\Delta\mathbf{R}$ are nonnegative, so we can define

$$\|\mathbf{N}\|_{1,\mathbf{V}} = \mathbf{N}^t\mathbf{V}, \qquad \|\mathbf{N}\|_{1,\boldsymbol{\Delta}\mathbf{R}} = \mathbf{N}^t\Delta\mathbf{R}.$$

As stated at the end of Section 8.1.2, we are going to use only a weighted version of the constraint (iii).

In order to insert the constraint (iii) we opportunely pad the matrix A and the right-hand side $\mathbf{I}$. Let $\gamma > 0$ be a fixed real number and define

$$\varphi = \frac{< A >_{i,j}}{< \mathbf{V} >_j},$$

where $< A >_{i,j}$ and $< \mathbf{V} >_j$ denote the arithmetic averages of the entries of the matrix A and the vector $\mathbf{V}$, respectively. We define

$$\tilde{A}_\gamma = \begin{pmatrix} A \\ \gamma\varphi\mathbf{V}^t \end{pmatrix} \qquad \text{and} \qquad \tilde{\mathbf{I}}_\gamma = \begin{pmatrix} \mathbf{I} \\ \gamma\varphi c_V \end{pmatrix}.$$

Note that, since the factor $\gamma\varphi V_j = \gamma < A >_{i,j} (V_j / < \mathbf{V} >_j)$ appearing in (8.17), must have the same dimensional units as $< A >_{i,j}$, γ necessarily has to be a dimensionless parameter and, therefore, its effect on (8.17) is independent of the units of both $A_{i,j}$ and V_j.

In this way we have inserted the constraint (iii) in our system weighted by the value γ. The value of γ determines the strength of the constraint and its effectiveness. In particular, the larger γ the stronger the effect of the constraint will be.

The new extended system becomes

$$\tilde{A}_\gamma\mathbf{N} = \begin{pmatrix} A \\ \gamma\varphi\mathbf{V}^t \end{pmatrix}\mathbf{N} = \tilde{\mathbf{I}}_\gamma = \begin{pmatrix} \mathbf{I} \\ \gamma\varphi c_V \end{pmatrix} \tag{8.13}$$

Since, the entries of the matrix $\tilde{A}$ and $\tilde{\mathbf{I}}$ are nonnegative, and, according to (i), we are looking for a nonnegative solution of (8.13), the LR iterative method, appears to be an excellent candidate for undertaking this task. Indeed, provided that the initial guess $\mathbf{N}^0 > 0$, the $k+1$ approximated solution of equation(8.13) can be recursively written in term of the kth iterate as

$$\mathbf{N}^{k+1} = \frac{\mathbf{N}^k}{\mathbf{a}} \circ \left(\tilde{A}_\gamma^t \cdot \frac{\tilde{\mathbf{I}}_\gamma}{\tilde{\mathbf{I}}_\gamma^k} \right), \qquad k = 0, 1, \ldots,$$

where $\frac{\bullet}{\bullet}$ and $\bullet \circ \bullet$ are the entry-wise division and multiplication, respectively, and $\bullet \cdot \bullet$ is the usual matrix-vector multiplication. The vector $\mathbf{a} \in \mathbb{R}^m$ is defined as

$$a_j = \sum_{i=1}^{n+1} \left(\tilde{A}_\gamma^t \right)_{j,i} = \sum_{i=1}^{n} A_{i,j} + \gamma\varphi V_j, \qquad j = 1, \ldots, m, \tag{8.14}$$

and $\tilde{\mathbf{I}}_\gamma^k$ is

$$\tilde{\mathbf{I}}_\gamma^k = \tilde{A}_\gamma \mathbf{N}^k = \begin{pmatrix} A\mathbf{N}^k \\ \gamma\varphi \left\|\mathbf{N}^k\right\|_{1,\mathbf{V}} \end{pmatrix}.$$

Let us consider the $j-th$ component of $\mathbf{N}^{k+1}$:

$$N_j^{k+1} = \frac{N_j^k}{a_j}\left[\tilde{A}_\gamma^t \cdot \frac{\tilde{\mathbf{I}}_\gamma}{\tilde{\mathbf{I}}_\gamma^k}\right]_j = \frac{N_j^k}{a_j}\xi_j, \tag{8.15}$$

the factor ξ_j is

$$\xi_j = \sum_{i=1}^{n+1}\left(\tilde{A}_\gamma^t\right)_{j,i}\frac{\left(\tilde{\mathbf{I}}_\gamma\right)_i}{\left(\tilde{\mathbf{I}}_\gamma^k\right)_i} = \sum_{i=1}^{n+1}\left(\tilde{A}_\gamma\right)_{i,j}\frac{\left(\tilde{\mathbf{I}}_\gamma\right)_i}{\left(\tilde{\mathbf{I}}_\gamma\right)_i^k} = \sum_{i=1}^{n} A_{i,j}\frac{I_i}{I_i^k} + \gamma\varphi V_j\frac{c_V}{c_V^k}, \tag{8.16}$$

with $c_V^k = \left\|\mathbf{N}^k\right\|_{1,\mathbf{V}}$. Combining (8.14) and (8.16) with (8.15), we obtain

$$N_j^{k+1} = N_j^k\frac{\sum_{i=1}^n A_{i,j}\frac{I_i}{I_i^k} + \gamma\varphi V_j\frac{c_V}{c_V^k}}{\sum_{i=1}^n A_{i,j} + \gamma\varphi V_j}, \qquad j = 1,\ldots,m, \tag{8.17}$$

where we have called $\mathbf{I}^k = A\mathbf{N}^k$. We can see that constraint (iii) is not blended with the other term, it is decoupled from the data fitting part and is weighted by γ. Moreover, the nonnegativity of $\mathbf{N}^k$ is simply preserved starting with $\mathbf{N}^0 > 0$, e.g., $\mathbf{N}^0 = \mathbf{1}$, where $\mathbf{1}$ represents the vector with entries all equals to 1.

8.2.1 Heuristic interpretation

We now want to give an heuristic interpretation of the formulation of (8.13).

A standard approach for passing from a constrained least square problem to an unconstrained problem is the well-known quadratic penalization technique [16]. In our case, considering the constraint (iii), we obtain the minimization problem

$$\min_{\mathbf{N}} \|A\mathbf{N} - \mathbf{I}\|^2 + (\gamma\varphi)^2\left(\|\mathbf{N}\|_{1,\mathbf{V}} - c_V\right)^2,$$

where $(\gamma\varphi)^2\left(\|\mathbf{N}\|_{1,\mathbf{V}} - c_V\right)^2$ is the penalization term. This can be seen as a regularized version of the problem (8.2), where the parameter $(\gamma\varphi)^2$ balances the trade off between the data fitting and the penalization term.

Define

$$\begin{aligned}\Psi(\mathbf{N}) &= \|A\mathbf{N} - \mathbf{I}\|^2 + (\gamma\varphi)^2\left(\|\mathbf{N}\|_{1,\mathbf{V}} - c_V\right)^2 \\ &= \mathbf{N}^t A^t A\mathbf{N} - 2\mathbf{N}^t A^t\mathbf{I} + \mathbf{I}^t\mathbf{I} + (\gamma\varphi)^2\left(\mathbf{N}^t\mathbf{V}\mathbf{V}^t\mathbf{N} - 2c_V\mathbf{N}^t\mathbf{V} + c_V^2\right).\end{aligned}$$

The gradient of $\Psi(\mathbf{N})$ is

$$\nabla\Psi(\mathbf{N}) = 2[A^tA\mathbf{N} - A^t\mathbf{I} + (\gamma\varphi)^2(\mathbf{V}\mathbf{V}^t\mathbf{N} - c_V\mathbf{V})].$$

Assume that $V_j \neq 0$ for all j. Then $\Psi(\mathbf{N})$ is coercive. In fact

$$\Psi(\mathbf{N}) \geq \gamma\varphi \left(\min_j \{V_j\} \mathbf{N}^t \mathbf{1} - c_V \right)^2 \to \infty \text{ as } \|\mathbf{N}\| \to \infty.$$

Thus the minimum of Ψ satisfies

$$\nabla \Psi(\mathbf{N}) = 0. \tag{8.18}$$

Condition (8.18) can be rewritten equivalently as

$$\begin{pmatrix} A^t & \gamma\varphi \mathbf{V} \end{pmatrix} \begin{pmatrix} A \\ \gamma\varphi \mathbf{V}^t \end{pmatrix} \mathbf{N} = \begin{pmatrix} A^t & \gamma\varphi \mathbf{V} \end{pmatrix} \begin{pmatrix} \mathbf{I} \\ \gamma\varphi c_V \end{pmatrix}, \tag{8.19}$$

Recalling the definitions of $\tilde{A}_\gamma$ and $\tilde{\mathbf{I}}_\gamma$, we can see that (8.19) is simply the normal equations of (8.13). This, coupled with (i), leads to the idea of looking for a nonnegative solution of (8.13).

8.2.2 Estimation of γ

The parameter γ weights the constraint (iii), but larger values of γ not necessarily lead to better restorations in practice (see Figures 8.4 and 8.6). This shows that the use of a constrained optimization algorithm does not necessarily provide better reconstructions.

The estimation of the optimal value of γ, i.e., the one that minimizes the reconstruction error, can be somewhat tricky and the choice of γ is not straightforward, thus we propose to use an a posteriori strategy using the constraint (ii).

Let us call $\mathbf{N}_\gamma$ the nonnegative solution of (8.19) obtained with a certain choice of γ and suppose to know exactly c_N. Thus, we expect that the best choice for γ is the one that, beside providing the best reconstruction for $\mathbf{N}_\gamma$, minimizes also the error on c_N. Therefore, we choose $\gamma = \gamma_{\text{opt}}$ such that

$$\gamma_{\text{opt}} = \arg\min_\gamma \{ |c_N - \|\mathbf{N}_\gamma\|_{1,\mathbf{\Delta R}}| \}. \tag{8.20}$$

In practice, as we will see in Section 8.3, the rule choice (8.20) is not so strict because there is a large range of γ-values around γ_{opt} where the reconstruction is equally good and even a value of γ very far from γ_{opt} would provide accurate results. This feature is of fundamental importance because, whenever the constraint c_N is not known and can be only roughly estimated (as it might happen experimentally), γ_{opt} cannot be determined with high accuracy and the condition (8.20) would be inapplicable.

Summarizing, our *weakly constrained LR (WCLR)* algorithm is the following:

1. fix $\mathbf{N}^0 = \mathbf{1}$ and a small set of possible values for γ;
2. compute $\mathbf{N}_\gamma = \mathbf{N}^k$, with k large enough, by (8.17) for every γ;
3. choose the solution $\mathbf{N}_{\gamma_{\text{opt}}}$ corresponding to γ_{opt} defined in (8.20).

8.3 Numerical Examples

In this section we report some numerical examples aimed at ascertaining how our algorithm performs against the classical LR method, i.e., when $\gamma = 0$. We will also show how to find the optimal value γ_{opt}, consistently with what described in Subsection 8.2.2.

Generation of matrix A

The $n \times m$ matrix A was computed by numerically integrating equation (8.4) with the kernel K provided by the Mie theory [91] and illustrated in Figure 8.1. The number of angles was $n = 100$, scaled according to a geometrical progression with $\theta_{\min} = 2\,deg$ and $\theta_{\max} = 180\,deg$. The bins for the recovered distribution were also chosen accordingly to a geometrical progression (see Section 8.1.1) with $R_{\min} = 10^{-3}\,\mu m$ and $R_{\max} = 10^{3}\,\mu m$ and their number was $m = 600$. In this way all the bins were characterized by the same relative width $\Delta R_j / R_j \approx 0.023$.

Generation of artificial test data

The generation of artificial test data was carried out by supposing to know a true solution for the system $\mathbf{N}_{\text{true}}$ and computing the noise-free data as

$$\mathbf{I}_{\text{true}} = A\mathbf{N}_{\text{true}}.$$

Then the real data were obtained by adding to $\mathbf{I}_{\text{true}}$ a (fractional) random white Gaussian noise, so that the noise level is proportional to $\mathbf{I}_{\text{true}}$ and independent from point to point. If we indicate with ϵ the fractional noise level, the noisy $\mathbf{I}$ becomes

$$\mathbf{I} = \mathbf{I}_{\text{true}} \circ (\mathbf{1} + \epsilon\mathbf{e}), \tag{8.21}$$

where $\mathbf{e}$ is a vector whose entries are realizations of a random variable such that $(\mathbf{e})_j \sim \mathcal{N}(0,1)$. Typical values for ϵ were 10^{-3} to 10^{-2}.

Inversion and parameters evaluation procedures

The artificial test data in (8.21) were inverted by using (8.17) with different values of the parameter γ, and the accuracy of the inversion algorithm was evaluated by comparing the retrieved distribution with the true one. However, since from a physical point of view, volume (or mass) distributions are much more significant than number distributions, we compared retrieved and true distributions on the basis of volume-fraction density distributions, defined as

$$\phi(R) = N(R)v(R),$$

where $\phi(R)$ has the dimensions of $[\mu m^{-1}]$.

For assessing the accuracy of the inversion procedure, we define a γ-dependent RRE as

$$RRE(\gamma) = \frac{\|\phi_\gamma - \phi_{\text{true}}\|}{\|\phi_{\text{true}}\|},$$

which corresponds to the relative average root mean square deviations between the retrieved and true mass distributions.

Similarly, for assessing the accuracy on the recovered values of the two parameters c_N and c_V which characterize the true distribution, we define the quantities

$$D_N(\gamma) = \frac{\left| \|\mathbf{N}_\gamma\|_{1,\mathbf{\Delta R}} - c_N \right|}{c_N} \qquad \text{and} \qquad D_V(\gamma) = \frac{\left| \|\mathbf{N}_\gamma\|_{1,\mathbf{V}} - c_V \right|}{c_V},$$

which represent the relative errors between c_N and c_V and the corresponding recovered parameters.

Numerical results

In the following numerical tests, the WCLR and LR algorithms were stopped after 10^6 iterations, beyond which the recovered distribution did not changed anymore. The approximate solution was computed for several values of γ in a given range of recoverable radii, which was selected as a two order of magnitude large subset ($[R_{\max}/R_{\min} = 100]$) of the original range defined above. For each γ, the tests were repeated 30 times, with different noise realizations (of the same ϵ level).

Test 1 In the first test we show that, when the distribution to be recovered is characterized by particles that produce signals whose features are: (a) asymptotically constant at low angles and (b) exhibit a dynamic range between first and last angle of several orders of magnitude [see $d = 0.1\,\mu m$ or $d = 1.0\,\mu m$ curves in Figure 8.1(a)], the original LR and our WCLR algorithms are quite equivalent. To this aim, we selected as true number distribution $\mathbf{I}_{\text{true}}$ a Gaussian centered in the middle of the recoverable range $[R_{\min},\, R_{\max}]$, i.e., with an average value $\langle R \rangle = 1\,\mu m$ and standard deviation $\sigma_R = 0.1\,\mu m$. The particle number concentration was (arbitrarily) chosen to be $c_N = 10^{16}\,cm^{-3}$ and the $r.m.s.$ noise level added to the data was $\epsilon = 0.01$. The inversion was carried out by trimming the recoverable particle radii in the range $[0.1\,\mu m - 10\,\mu m]$ (so that the number of bins was $m = 200$).

The value of γ ranges from zero (original LR) to $\gamma = 10^6$. The findings of this test show that our algorithm performs equally well independently of the $\gamma-$value ($0 - 10^4$) and its performances were quite similar to the ones provided by the original LR algorithm. This is shown in Figure 8.3 where data reconstructions and average recovered distributions are highly accurate and, as matter of fact, indistinguishable between our algorithm (run with $\gamma = 1$) and the classical LR algorithm ($\gamma = 0$).

Test 2 The effective difference between the two algorithms becomes evident only when the particles are close to the sides of the $[R_{\min},\, R_{\max}]$ range. For this second test we selected a Gaussian distribution characterized by large particles, i.e., with $\langle R \rangle = 100\,\mu m$ and $\sigma_R = 10\,\mu m$. The inversion was carried out in the range $[10\,\mu m - 1000\,\mu m]$ varying γ in $[10^{-2} - 10^7]$. Differently from the first test, the effect of changing γ is quite relevant, as shown in Figure 8.4 where the behaviors of the parameters RRE (a), D_N (b), and D_V (c) are reported as a function of γ. As one can notice, whereas D_V decreases monotonically with increasing γ (which is consistent with the fact that the stronger the constraint, the higher the accuracy of its recovered value), the other two parameters exhibit very broad valleys whose flat regions cover almost the same range of $\gamma \sim [10^1 - 10^5]$. Thus, the choice of an optimal value for

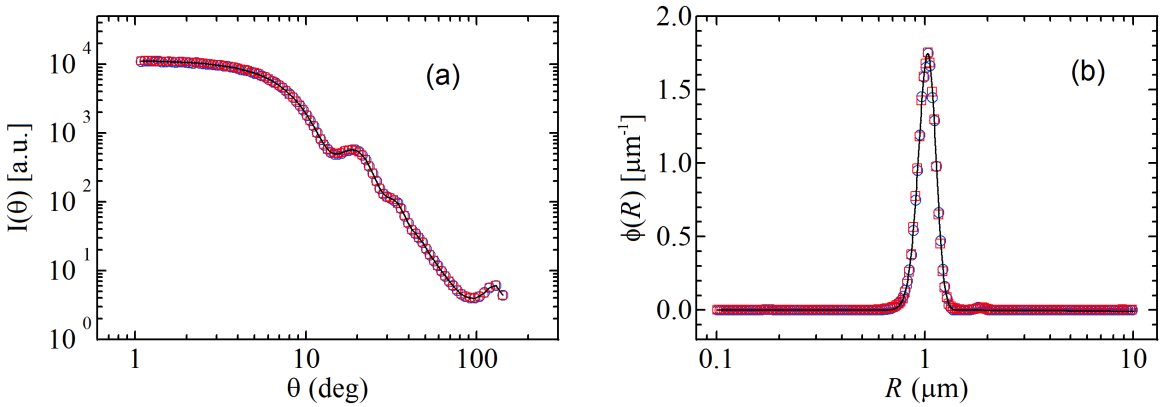


FIGURE 8.3: Test 1: Comparison between the original LR (blue symbols) and our WCLR algorithm (red symbols) run with $\gamma = 1$ for a Gaussian distribution with an average value $\langle R \rangle = 1\,\mu m$ and standard deviation $\sigma_R = 0.1\,\mu m$. (a) Reconstructed (symbols) and true (line) signals. (b) Reconstructed (symbols) and true (line) distributions. The two algorithms performs equally well and, as matter of fact, are almost indistinguishable results.

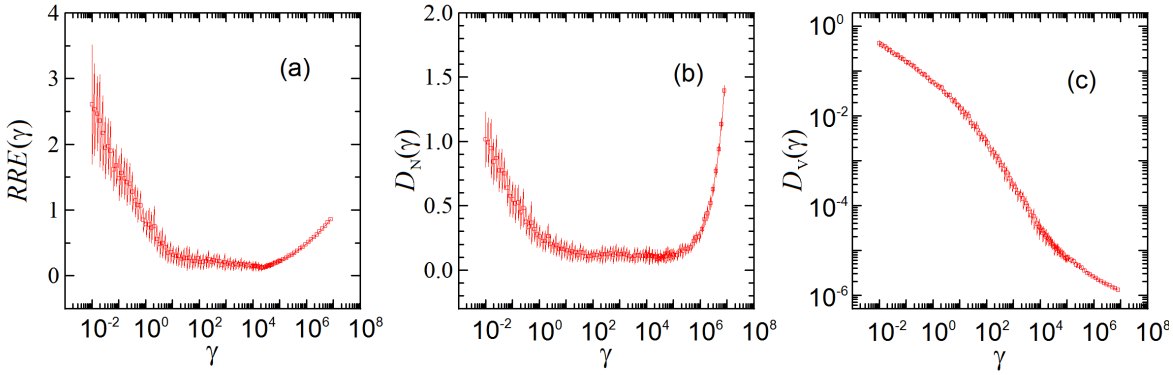


FIGURE 8.4: Test 2: Behavior as a function of γ of the average parameters RRE (a), D_N (b), and D_V (c) for a Gaussian distribution with $\langle R \rangle = 100\,\mu m$ and $\sigma_R = 10\,\mu m$. The error bars are the standard deviations associated to the various parameters deriving from the noise ($\epsilon = 0.01$) present on the data.

γ is not critical at all and any value chosen in the central part of this interval (for example $\gamma_{\text{opt}} \sim [10^2 - 10^4]$) leads both to small errors in the recovery of c_N and to very accurate distribution reconstructions, as shown in panels Figure 8.5(e-f-g). Conversely, for values of γ outside this range, the recovery of c_N becomes increasingly less and less accurate and, at the same time, the distributions are recovered more and more poorly, as shown in all the other panels of the Figure 8.5. In particular, we would like to point out the remarkable mismatching between the true distribution and the one recovered in Figure 8.5(a), showing that the classical LR algorithm ($\gamma = 0$) is totally unable to perform such a task. Finally, we would like to point out that the rather similar behaviors between RRE and D_N guarantees that, in a real experiment where the parameter RRE cannot be measured because $\phi_{\text{true}}(R)$ is not known, the optimal range for γ_{opt} can be inferred by looking at the behavior of D_N (Figure 8.4(b)). The fact that such a range is remarkably broad ($\sim$ 3 orders of magnitude) ensures that even a huge uncertainty on the value of c_N would not affect significantly the accuracy with which $\phi_{\text{true}}(R)$ will be reconstructed.

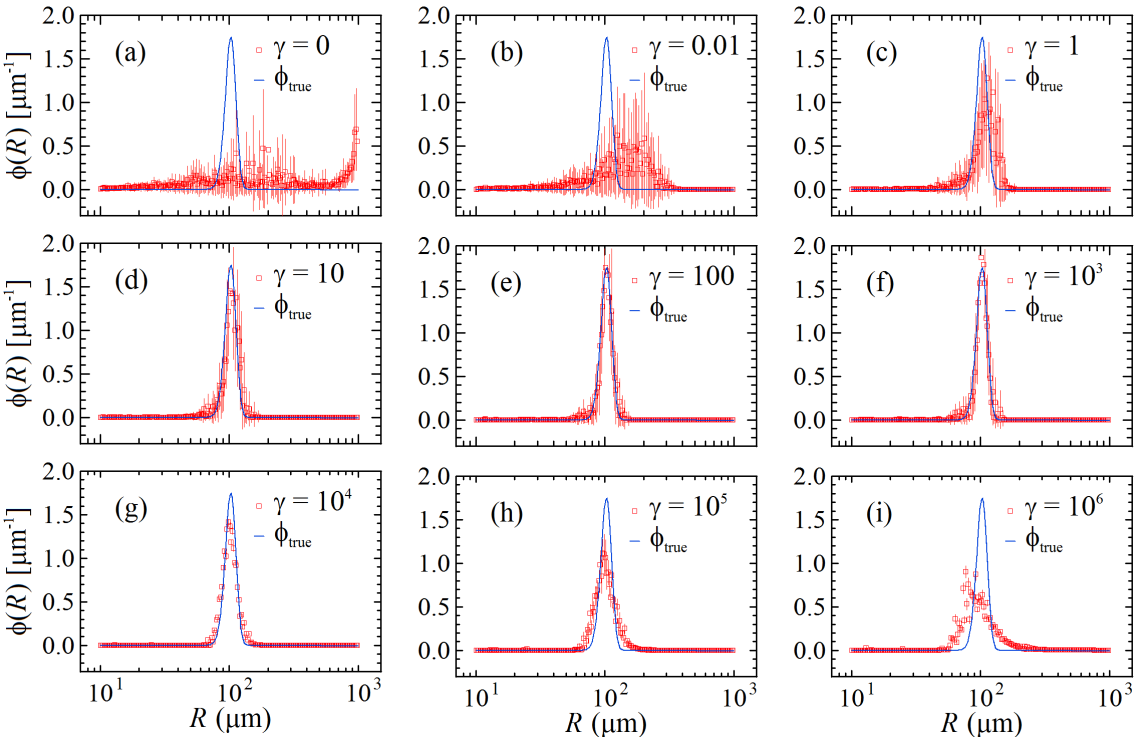


FIGURE 8.5: Test 2: Comparison between the average recovered distributions (red symbols) obtained with our algorithm at various γ-values ($\gamma = 0$ is LR) and a true Gaussian distribution $\phi_{\text{true}}(R)$ with with $\langle R \rangle = 100\,\mu m$ and $\sigma_R = 10\,\mu m$. The error bars are the standard deviations associated to the bins of the recovered histogram due to the noise (a $\epsilon = 0.01$) present on the data.

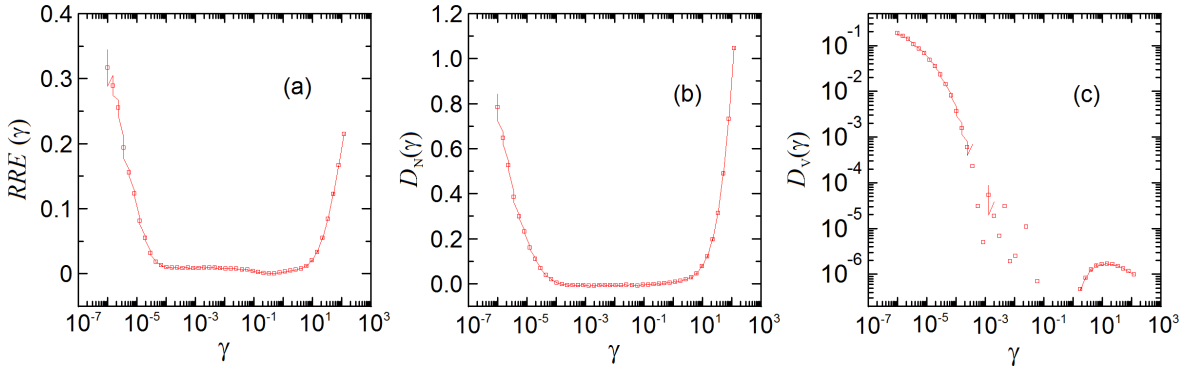


FIGURE 8.6: Test 3: Behavior as a function of γ of the average parameter RRE (a), D_N (b), and D_V (c) for a Gaussian distribution with $\langle R \rangle = 0.05\,\mu m$ and $\sigma_R = 0.005\,\mu m$ with $\langle R \rangle = 100\,\mu m$ and $\sigma_R = 10\,\mu m$. The error bars are the standard deviations associated to the various parameters deriving from the noise ($\epsilon = 0.001$) present on the data.

Test 3 As a final test, we selected a distribution close to the left side of the range, namely a Gaussian with $\langle R \rangle = 0.05\,\mu m$ and $\sigma_R = 0.005\,\mu m$. In this case, as shown in Figure 8.1, the $I(\theta)$ data are rather flat and therefore, for not corrupting completely their behavior, the noise level added to the data was $\epsilon = 0.001$. All the other inversion parameters were identical to the ones used in the previous test, except for the range of recoverable radii that was $[0.005\,\mu m - 0.5\,\mu m]$. The behaviors of the parameters RRE (a), D_N (b), and D_V (c) are reported in Figure 8.6, whereas the comparison between the recovered distribution and $\phi_{\text{true}}(R)$ is shown in the nine panels of Figure 8.7 varying γ in the range $[10^{-6} - 10^{3}]$. As for the large particles, there is an optimal range $\gamma_{\text{opt}} \sim [10^{-4} - 10^{1}]$ where both distribution reconstruction and c_N recovery are very accurate. Also in this case, the original LR algorithm is totally unable to recover the true distribution (see Figure 8.7(a)). And finally, given the similar behaviors of the parameters of RRE and D_N which exhibits very broad and shallow minima, the same method described above for the estimate of the optimal range for γ_{opt} can be applied.

To conclude this section we observe that in Figure 8.7(c) we can see a strange behavior of the quantity D_V with respect to γ. In particular, we would expect this quantity to be monotonically decreasing as γ increases, however, this is not the case. What happens in this scenario is that when γ is very large, the constraint is very effective thus slowing down the convergence of method. The "bumb" that can be seen in Figure 8.7(c) is due to the fact that the maximum number of iteration was reached before convergence. In order to solve this issue we plan to insert a more sophisticated stopping criterion, related to the discrepancy principle, that will avoid this problem.

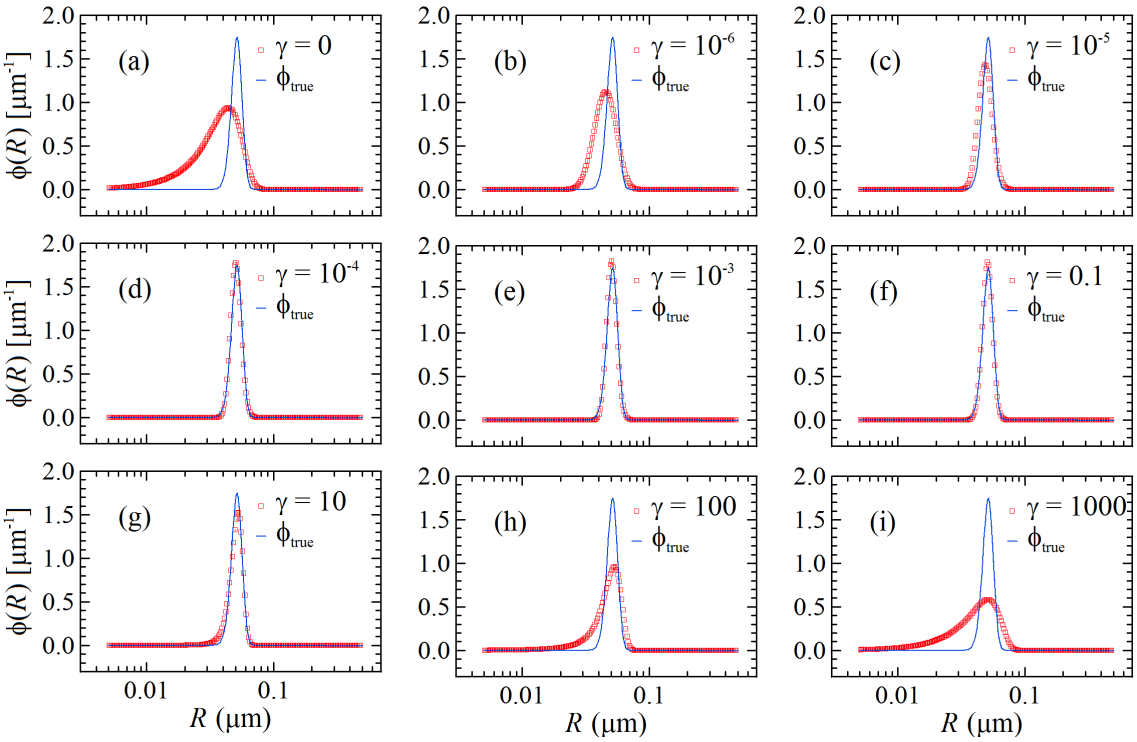


FIGURE 8.7: Test 3: Comparison between the average recovered distributions (red symbols) obtained with our algorithm at various γ-values ($\gamma = 0$ is LR) and a true Gaussian distribution $\phi_{\text{true}}(R)$ with with $\langle R \rangle = 0.05\,\mu m$ and $\sigma_R = 0.005\,\mu m$. The error bars are the standard deviations associated to the bins of the recovered histogram due to the noise (a $\epsilon = 0.001$) present on the data.

Chapter 9

A semi-blind regularization algorithm for inverse problems with application to image deblurring

In this last chapter we consider inverse problems in Sobolev spaces which can be modeled as an equation of the form

$$B(k, f) = g, \tag{9.1}$$

where f is the desired solution, g is the measured data, and k is a variable on which the operator B depends, e.g., if $B(k, f) = k * f$ is the convolution operator, then k represents the integral kernel.

We consider the situation in which (9.1) is ill-posed and both k and g are corrupted by noise. We denote by k_ϵ and g_δ the noise corrupted version of k and g, respectively, and we assume that the following bounds hold

$$\|g - g_\delta\| \leq \delta \qquad \text{and} \qquad \|k - k_\epsilon\| \leq \epsilon.$$

Therefore, equation (9.1) becomes

$$B(k_\epsilon, f) = g_\delta.$$

We refer to the regularization of such a problem as *semi-blind regularization*, since the variable k, even though it is not completely unknown, has a certain degree of uncertainty on it.

Blind and semi-blind deconvolution has been widely investigated [1, 11, 17, 18, 20, 39, 41, 51, 66, 85, 109]. The approach in [11, 51] requires that the blurring operator is diagonalized by fast transforms, which is not assumed in this chapter. In [17] a double regularization approach to recover f and k was proposed, which consisted in solving

$$\arg\min_{k,f} J(k, f) \tag{9.2}$$

where

$$J(k, f) = \frac{1}{2} T(k, f) + R(k, f),$$

T had the role of data-fitting term and R was the penalty term. In particular,

$$\begin{aligned} T(k, f) &= \|B(k, f) - g_\delta\|^2 + \gamma \|k - k_\epsilon\|^2, \\ R(k, f) &= \alpha \|Lf\| + \beta \mathcal{R}(k), \end{aligned}$$

where $\mathcal{R}(k)$ is an appropriate penalty term and L is a regularization operator which is bounded and continuously invertible. A possible numerical technique to solve (9.2) was proposed by the same authors in [18] using an alternating minimization over f and k.

In this chapter we want to improve the results in [17, 18] by introducing a more complex penalty term for f and by constrain the minimization. In particular, we want to introduce, for both k and f, the *Total Variation* (TV) as a prior and add nonnegativity and flux conservation constraints. The introduction of TV and the constraints complicates the minimization of the resulting functional, thus the usage of advanced numerical techniques is required.

Summarizing, the problem that we would like to solve is

$$\left(k_{\alpha,\beta}^{\delta,\epsilon}, f_{\alpha,\beta}^{\delta,\epsilon}\right) = \arg \min_{(k,f)\in\Omega_k\times\Omega_f} J_{\alpha,\beta}^{\delta,\epsilon}(k,f) \tag{9.3}$$

where $J_{\alpha,\beta}^{\delta,\epsilon}$ is the non-smooth and non-convex functional defined in (9.4).

Firstly we prove some theoretical properties of $J_{\alpha,\beta}^{\delta,\epsilon}$. In particular, we show the existence of a global minimum, the stability property and that, if α and β are chosen properly, (9.3) is a regularization method.

Then, for the numerical solution of (9.3), we use the well known *Alternating Direction Multiplier Method* (ADMM). However, since $J_{\alpha,\beta}^{\delta,\epsilon}$ is non-convex, the classical theory of ADMM does not assure the convergence of the algorithm. Thus, we prove that ADMM applied to (9.3) converges to a stationary point of $J_{\alpha,\beta}^{\delta,\epsilon}$.

Finally, we give some numerical evidences of the improvement in term of quality of the reconstructed images with respect to the proposal in [18]. Moreover, we show that inserting k as a variable inside the model leads to more accurate approximations than using the measured k_ϵ directly, i.e., than solving

$$f_{\alpha,\beta}^{\delta,\epsilon} = \arg \min_{f\in\Omega_f} J_{\alpha,\beta}^{\delta,\epsilon}(k_\epsilon, f).$$

This chapter is structured as follows. In Section 9.1 we describe the functional we consider and analyze some of its properties. In Section 9.2 we discuss the addition of some constraints, while in Section 9.3 we describe the algorithm for the minimization of the functional previously introduced and we prove its convergence. Section 9.4 is devoted to numerical examples in image deblurring.

9.1 The regularized functional

As previously discussed, our goal is to extend the results form [17] using a more complex penalty term obtained by considering the TV for both f and k. Let $g_\delta, k_\epsilon, f, k \in H^1$, where H^1 denotes the Sobolev space $W^{1,2}$ which is a separable Hilbert space, see, e.g., [23, Section 9.1]. We consider the minimization of the following functional

$$J_{\alpha,\beta}^{\delta,\epsilon}(k,f) = \|B(k,f) - g_\delta\|^2 + \alpha\left(\|f\|_{TV} + \|f\|^2\right) + \gamma\|k - k_\epsilon\|^2 + \beta\|k\|_{TV}, \tag{9.4}$$

where

$$\|h\|_{TV} = \int \|\nabla h\|$$

is the total variation of $h \in H^1$. The penalty term on f is able to ensure a certain degree of smoothness, thanks to the $\|\cdot\|$ term, while preserving edges at the same time, using the $\|\cdot\|_{TV}$ term.

We now show some theoretical properties of (9.4), in particular, we want to show the existence of a global minimizer, the stability property, and the fact that, if α and β are chosen in relation to the noise, the minimization of $J^{\delta,\epsilon}_{\alpha,\beta}$ induces a regularization technique.

We assume that

Assumption 9.1. *The operator B is strongly continuous on its domain.*

Before proving the existence of the minimizer we first need to show some properties of $J^{\delta,\epsilon}_{\alpha,\beta}(f,k)$.

Lemma 9.1. *The functional $J^{\delta,\epsilon}_{\alpha,\beta}(f,k)$ defined in* (9.4) *is positive, weakly lower semi-continuous (wlsc) and coercive with respect to the norm $\|(k,f)\|^2 := \|k\|^2 + \|f\|^2$.*

Proof. It is obvious that $J^{\delta,\epsilon}_{\alpha,\beta}$ is positive and wlsc since it is a sum of positive and wlsc functions. The coercivity trivially follows from

$$J^{\delta,\epsilon}_{\alpha,\beta}(k,f) \geq \gamma \|k - k_\epsilon\|^2 + \alpha \|f\|^2 \to \infty \text{ as } \|(k,f)\| \to \infty.$$

□

We are now able to prove the existence of a global minimizer.

Theorem 9.2 (Existence)**.** *The functional $J^{\delta,\epsilon}_{\alpha,\beta}(f,k)$ defined in* (9.4) *has a global minimizer.*

Proof. From Lemma 9.1 we know that $J^{\delta,\epsilon}_{\alpha,\beta}$ is positive, proper, and coercive, thus $\exists\,(k,f) \in \mathcal{D}\left(J^{\delta,\epsilon}_{\alpha,\beta}\right)$, where $\mathcal{D}\left(J^{\delta,\epsilon}_{\alpha,\beta}\right)$ denotes the domain of $J^{\delta,\epsilon}_{\alpha,\beta}$, such that $J^{\delta,\epsilon}_{\alpha,\beta}(k,f) < \infty$. Let us call

$$\nu := \inf\left\{J^{\delta,\epsilon}_{\alpha,\beta}(k,f) : (k,f) \in \mathcal{D}\left(J^{\delta,\epsilon}_{\alpha,\beta}\right)\right\}, \tag{9.5}$$

we want to show that ν is attained, meaning that the infimum is actually a minimum.

By definition of ν there exist $M > 0$ and $\{(k_j,f_j)\}_j \in \mathcal{D}\left(J^{\delta,\epsilon}_{\alpha,\beta}\right)$ such that

$$J^{\delta,\epsilon}_{\alpha,\beta}(k_j,f_j) \to \nu \quad \text{and} \quad J^{\delta,\epsilon}_{\alpha,\beta}(k_j,f_j) \leq M \;\forall j. \tag{9.6}$$

From (9.6) we get that $\alpha\|f_j\|^2 \leq M$ and $\gamma\|k_j - k_\epsilon\|^2 \leq M$, moreover,

$$\|k_j\| - \|k_\epsilon\| \leq \|k_j - k_\epsilon\| \leq \left(\frac{M}{\gamma}\right)^{\frac{1}{2}}.$$

Thus the following bounds hold

$$\|k_j\| \leq \left(\frac{M}{\gamma}\right)^{\frac{1}{2}} + \|k_\epsilon\| \quad \text{and} \quad \|f_j\| < \left(\frac{M}{\alpha}\right)^{\frac{1}{2}},$$

i.e., the sequence $\{(k_j,f_j)\}_j$ is uniformly bounded, so there exists a subsequence $\{(k_j,f_j)\}_j$ (with abuse of notation we use the same indexes) such that $k_j \rightharpoonup \bar{k}$ and $f_j \rightharpoonup \bar{f}$, i.e., $(k_j,f_j) \rightharpoonup (\bar{k},\bar{f})$.

We now prove that ν is the minimum of the functional $J^{\delta,\epsilon}_{\alpha,\beta}$ and is attained at $\left(\bar{k},\bar{f}\right)$, i.e., $\left(\bar{k},\bar{f}\right)$ is a global minimizer. By wlsc of $J^{\delta,\epsilon}_{\alpha,\beta}$ we have

$$\nu \leq J^{\delta,\epsilon}_{\alpha,\beta}\left(\bar{k},\bar{f}\right) \leq \liminf_{j\to\infty} J^{\delta,\epsilon}_{\alpha,\beta}\left(k_j,f_j\right) = \lim_{j\to\infty} J^{\delta,\epsilon}_{\alpha,\beta}\left(k_j,f_j\right) = \nu.$$

So $\nu = J^{\delta,\epsilon}_{\alpha,\beta}\left(\bar{k},\bar{f}\right)$ is the minimum of the functional and $\left(\bar{k},\bar{f}\right)$ is a global minimizer. □

We are now in a position to show the stability property

Theorem 9.3 (Stability)**.** *With the same notation as above, let α and β be fixed. Let $\left\{\left(g_{\delta_j}\right)\right\}_j$ and $\left\{\left(k_{\epsilon_j}\right)\right\}_j$ be sequences such that $g_{\delta_j} \to g_\delta$ and $k_{\epsilon_j} \to k_\epsilon$, let $\{(k_j,f_j)\}_j$ be minimizers obtained with data $g_{\delta_j}, k_{\epsilon_j}$. Then there exists a convergent subsequence of $\{(k_j,f_j)\}_j$ and the limit of every subsequence is a minimizer of $J^{\delta,\epsilon}_{\alpha,\beta}$.*

Proof. Because $\{(k_j,f_j)\}_j$ are minimizers it holds that

$$J^{\delta_j,\epsilon_j}_{\alpha,\beta}\left(k_j,f_j\right) \leq J^{\delta_j,\epsilon_j}_{\alpha,\beta}\left(k,f\right) \qquad \forall\,(k,f) \in \mathcal{D}\left(J^{\delta_j,\epsilon_j}_{\alpha,\beta}\right). \tag{9.7}$$

Let us denote by $\left(\tilde{k},\tilde{f}\right)$ the minimizers of $J^{\delta,\epsilon}_{\alpha,\beta}$, i.e., $\left(\tilde{k},\tilde{f}\right) := \left(k^{\delta,\epsilon}_{\alpha,\beta}, f^{\delta,\epsilon}_{\alpha,\beta}\right)$.

Since $J^{\delta_j,\epsilon_j}_{\alpha,\beta}\left(\tilde{k},\tilde{f}\right) \to J^{\delta,\epsilon}_{\alpha,\beta}\left(\tilde{k},\tilde{f}\right)$ there exists $\tilde{c} > 0$ such that

$$J^{\delta_j,\epsilon_j}_{\alpha,\beta}\left(\tilde{k},\tilde{f}\right) \leq \tilde{c} \qquad \text{for } j \text{ large enough.}$$

The latter implies that $\{\|k_j - k_\epsilon\|\}_j$ and $\{\|f_j\|\}_j$ are uniformly bounded and so, like in Theorem 9.2, it holds that $\{(k_j,f_j)\}_j$ is uniformly bounded.

With abuse of notation there exists a subsequence $\{(k_j,f_j)\}_j$ such that

$$k_j \rightharpoonup \bar{k} \quad \text{and} \quad f_j \rightharpoonup \bar{f}.$$

By wlsc of B and of $\|\cdot\|$ we have

$$\left\|B\left(\bar{k},\bar{f}\right) - g_\delta\right\| \leq \liminf_{j\to\infty} \left\|B\left(k_j,f_j\right) - g_{\delta_j}\right\|$$

and

$$\left\|k_j - k_\epsilon\right\| \leq \liminf_{j\to\infty} \left\|k_j - k_{\epsilon_j}\right\|.$$

From (9.7) it derives

$$\begin{aligned} J^{\delta,\epsilon}_{\alpha,\beta}\left(\bar{k},\bar{f}\right) &\leq \liminf_{j\to\infty} J^{\delta_j,\epsilon_j}_{\alpha,\beta}\left(k_j,f_j\right) \leq \limsup_{j\to\infty} J^{\delta_j,\epsilon_j}_{\alpha,\beta}\left(k,f\right) \\ &= \lim_{j\to\infty} J^{\delta_j,\epsilon_j}_{\alpha,\beta}\left(k,f\right) = J^{\delta,\epsilon}_{\alpha,\beta}\left(k,f\right), \qquad \forall\,(k,f) \in \mathcal{D}\left(J^{\delta,\epsilon}_{\alpha,\beta}\right). \end{aligned}$$

In particular, $J^{\delta,\epsilon}_{\alpha,\beta}\left(\bar{k},\bar{f}\right) \leq J^{\delta,\epsilon}_{\alpha,\beta}\left(\tilde{k},\tilde{f}\right)$, but $\left(\tilde{k},\tilde{f}\right)$ is a minimizer and so $J^{\delta,\epsilon}_{\alpha,\beta}\left(\bar{k},\bar{f}\right) = J^{\delta,\epsilon}_{\alpha,\beta}\left(\tilde{k},\tilde{f}\right)$, implying that

$$\lim_{j\to\infty} J^{\delta_j,\epsilon_j}_{\alpha,\beta}\left(k_j,f_j\right) = J^{\delta,\epsilon}_{\alpha,\beta}\left(\bar{k},\bar{f}\right).$$

We have proven the weak convergence of k_j and f_j to $\bar{k}$ and $\bar{f}$, respectively. We now have to show that the convergence is strong. It is enough to prove that

$$\left\|\bar{k}\right\| \geq \limsup_{j\to\infty} \|k_j\| \quad \text{and} \quad \left\|\bar{f}\right\| \geq \limsup_{j\to\infty} \|f_j\| .$$

Let us suppose that $\exists\ \tau$ such that $\tau = \limsup_{j\to\infty} \|f_j\| > \left\|\bar{f}\right\|$. So there exists a subsequence $\{f_n\}_n$ of $\{f_j\}_j$ such that $f_n \rightharpoonup \bar{f}$ and $\|f_n\| \to \tau$.

$$\begin{aligned}
&\lim_{n\to\infty} \left(\|B(k_n, f_n) - g_{\delta_n}\|^2 + \alpha \|f_n\|_{TV} + \gamma \|k_n - k_{\epsilon_n}\|^2 + \beta \|k_n\|_{TV} \right) \\
&= \left\|B\left(\bar{k}, \bar{f}\right) - g_\delta\right\|^2 + \alpha \left\|\bar{f}\right\|_{TV} + \gamma \left\|\bar{k} - k_\epsilon\right\|^2 + \beta \left\|\bar{k}\right\|_{TV} + \alpha \left(\left\|\bar{f}\right\|^2 - \lim_{n\to\infty} \|f_n\|^2 \right) \\
&= \left\|B\left(\bar{k}, \bar{f}\right) - g_\delta\right\|^2 + \alpha \left\|\bar{f}\right\|_{TV} + \gamma \left\|\bar{k} - k_\epsilon\right\|^2 + \beta \left\|\bar{k}\right\|_{TV} + \alpha \left(\left\|\bar{f}\right\|^2 - \tau^2 \right) \\
&< \left\|B\left(\bar{k}, \bar{f}\right) - g_\delta\right\|^2 + \alpha \left\|\bar{f}\right\|_{TV} + \gamma \left\|\bar{k} - k_\epsilon\right\|^2 + \beta \left\|\bar{k}\right\|_{TV} ,
\end{aligned}$$

which contradicts the wlsc of B and the norms, thus $f_j \to \bar{f}$.

Similarly we prove that $k_j \to \bar{k}$.

Let us suppose that $\exists\ \tau$ such that $\tau = \limsup_j \|k_j - k_\epsilon\| > \left\|\bar{k} - k_\epsilon\right\|$. So there exists a subsequence $\{k_n\}_n$ of $\{k_j\}_j$ such that $k_n - k_\epsilon \rightharpoonup \bar{k} - k_\epsilon$ and $\|k_n - k_\epsilon\| \to \tau$. By triangular inequality

$$\|k_n - k_\epsilon\| - \|k_\epsilon - k_{\epsilon_n}\| \leq \|k_n - k_{\epsilon_n}\| \leq \|k_n - k_\epsilon\| + \|k_\epsilon - k_{\epsilon_n}\| ,$$

so

$$\lim_{n\to\infty} \|k_n - k_{\epsilon_n}\| = \lim_{n\to\infty} \|k_n - k_\epsilon\| .$$

Thus

$$\begin{aligned}
&\lim_{n\to\infty} \left(\|B(k_n, f_n) - g_{\delta_n}\|^2 + \beta \|k_n\|_{TV} \right) \\
&= \left\|B\left(\bar{k}, \bar{f}\right) - g_\delta\right\|^2 + \beta \left\|\bar{k}\right\|_{TV} + \beta \left(\gamma \left\|\bar{k} - k_\epsilon\right\|^2 - \lim_{n\to\infty} \|k_n - k_\epsilon\|^2 \right) \\
&= \left\|B\left(\bar{k}, \bar{f}\right) - g_\delta\right\|^2 + \beta \left\|\bar{k}\right\|_{TV} + \beta \left(\gamma \left\|\bar{k} - k_\epsilon\right\|^2 - \tau^2 \right) \\
&< \left\|B\left(\bar{k}, \bar{f}\right) - g_\delta\right\|^2 + \beta \left\|\bar{k}\right\|_{TV} ,
\end{aligned}$$

which contradicts the wlsc of B and the norms, so $k_j \to \bar{k}$. □

One of the most important properties of iterative regularization methods is the regularization property. We want to prove that in the noise-free case we can exactly recover an exact solution of the problem and in particular that with minimum norm. Moreover we want that, as the norm of the noise goes to 0, the corresponding reconstructions converge to the minimum norm solution of the problem as well. This intuitively means that, if there is not too much noise and if the parameters α and β are set accordingly, we can trust our method to give good approximation to the true solution.

In order to talk about the regularization property, we need to clearly define what is the *minimum norm solution* in this setup.

Definition 9.4. *The minimum norm solution of $B(k_0, f) = g_0$ is*

$$f^\dagger = \arg\min_{f\in H^1} \{\|f\|^2 + \|f\|_{TV} :\ B(k_0, f) = g_0\}.$$

We now prove that, if α and β are chosen properly and dependently from the noise, the method proposed is a regularization method.

Theorem 9.5 (Regularization property)**.** *Let $\left\{g_{\delta_j}\right\}_j$ and $\left\{k_{\epsilon_j}\right\}_j$ be sequences such that*

$$\left\|g_{\delta_j} - g_0\right\| < \delta \text{ and } \left\|k_{\epsilon_j} - k_0\right\| < \epsilon_j$$

and such that $\delta_j, \epsilon_j \to 0$ as $j \to \infty$. Let $\{\alpha\}_j$ and $\{\beta\}_j$ be sequences such that $\alpha_j, \beta_j \to 0$ as $j \to \infty$, moreover assume that it holds

$$\lim_{j\to\infty} \frac{\delta_j^2 + \gamma\epsilon_j^2}{\alpha_j} = 0 \quad \text{and} \quad \lim_{j\to\infty} \frac{\beta_j}{\alpha_j} = \eta \quad 0 < \eta < \infty.$$

Call $(k_j, f_j) := \left(k_{\alpha_j,\beta_j}^{\delta_j,\epsilon_j}, f_{\alpha_j,\beta_j}^{\delta_j,\epsilon_j}\right)$, then there exists a convergent subsequence of $\{(k_j, f_j)\}$ such that $k_j \to k_0$ and the limit of every convergent subsequence of f_j is the minimum norm solution.

Proof. Since (k_j, f_j) is a minimizer we have

$$J_{\alpha_j,\beta_j}^{\delta_j,\epsilon_j}(k_j, f_j) \le J_{\alpha_j,\beta_j}^{\delta_j,\epsilon_j}(k, f) \quad \forall (k, f) \in \mathcal{D}\left(J_{\alpha_j,\beta_j}^{\delta_j,\epsilon_j}\right),$$

in particular

$$0 \le J_{\alpha_j,\beta_j}^{\delta_j,\epsilon_j}(k_j, f_j) \le J_{\alpha_j,\beta_j}^{\delta_j,\epsilon_j}\left(k_0, f^\dagger\right) \le \delta_j^2 + \gamma_j\epsilon_j^2 + \alpha_j\left(\left\|f^\dagger\right\|_{TV} + \left\|f^\dagger\right\|^2\right) + \beta_j \left\|k_0\right\|_{TV},$$

so $\left\|B(k_j, f_j) - g_{\delta_j}\right\|^2$, $\left\|k_j - k_{\epsilon_j}\right\|^2$, $\|f_j\|^2$, $\|f_j\|_{TV}$, and $\|k_j\|_{TV}$ are uniformly bounded.

There exists a subsequence $\{(k_n, f_n)\}_n$ of $\{(k_j, f_j)\}_j$ such that $(k_n, f_n) \rightharpoonup \left(\bar{k}, \bar{f}\right)$. We want to show that $\bar{k} = k_0$ and that $\bar{f}$ is the minimum norm solution. Moreover, we want to prove that the convergence is strong.

Let us firstly show that $\bar{k} = k_0$. Indeed, it holds

$$\begin{aligned}
0 &\le \left\|B\left(\bar{k}, \bar{f}\right) - g_0\right\|^2 + \gamma\left\|\bar{k} - k_0\right\|^2 \\
&\le \liminf_{n\to\infty} \left\|B(k_n, f_n) - g_{\delta_n}\right\|^2 + \gamma_n \left\|k_n - k_{\epsilon_n}\right\|^2 \\
&\le \liminf_{n\to\infty} \delta_n^2 + \gamma_n\epsilon_n^2 + \alpha_n\left(\left\|f^\dagger\right\|^2 + \left\|f^\dagger\right\|_{TV}\right) + \beta_n \left\|k_0\right\|_{TV} \\
&= 0,
\end{aligned}$$

thus $\bar{k} = k_0$ and $B\left(\bar{k}, \bar{f}\right) = g_0$.

We now show that $\bar{f}$ is the minimum norm solution. We have that

$$\|f_n\|^2 + \|f_n\|_{TV} + \frac{\beta_n}{\alpha_n}\|k_n\|_{TV} \le \frac{\delta_n^2 + \gamma_n}{\alpha_n} + \left\|f^\dagger\right\|^2 + \left\|f^\dagger\right\|_{TV} + \frac{\beta_n}{\alpha_n}\|k_0\|_{TV}.$$

We get

$$\begin{aligned}\left\|\bar{f}\right\|^2+\left\|\bar{f}\right\|_{TV}+\eta\left\|\bar{k}\right\|_{TV} &\leq \liminf_{n\to\infty}\left(\|f_n\|^2+\|f_n\|_{TV}+\eta\|k_n\|_{TV}\right)\\ &=\liminf_{n\to\infty}\left(\|f_n\|^2+\|f_n\|_{TV}+\frac{\beta_n}{\alpha_n}\|k_n\|_{TV}\right)\\ &\leq \liminf_{n\to\infty}\left(\frac{\delta_n^2+\gamma_n}{\alpha_n}+\left\|f^\dagger\right\|^2+\left\|f^\dagger\right\|_{TV}+\frac{\beta_n}{\alpha_n}\|k_0\|_{TV}\right)\\ &=\left\|f^\dagger\right\|^2+\left\|f^\dagger\right\|_{TV}+\eta\|k_0\|_{TV},\end{aligned}$$

but $\bar{k}=k_0$ and so $\bar{f}$ is the minimum norm solution.

We finally prove that $f_n\to f^\dagger$ and $k_n\to k_0$.

We start with f_n, it is sufficient to show that $\|f_n\|\to\left\|f^\dagger\right\|$ or equivalently (by wlsc of the norm) that $\limsup_{n\to\infty}\|f_n\|\leq\left\|\bar{f}\right\|$. Let us suppose that there exists τ such that $\tau=\limsup_{n\to\infty}\|f_n\|^2>\left\|\bar{f}\right\|^2$ and so there is a subsequence $\{f_l\}_l$ of $\{f_n\}_n$ such that $f_l\rightharpoonup\bar{f}$ and $\|f_l\|^2\to\tau$, so

$$\limsup_{l\to\infty}\frac{\beta_l}{\alpha_l}\|k_l\|_{TV}=\eta\|k_0\|_{TV}+\left(\left\|\bar{f}\right\|^2-\limsup_{l\to\infty}\|f_l\|^2\right)<\eta\|k_0\|_{TV},$$

which is a contradiction to the wlsc of the norm. So we have that $f_n\to\bar{f}$.

As for k_n we have

$$\|k_n-k_0\|\leq\|k_n-k_{\epsilon_n}\|+\|k_{\epsilon_n}-k_0\|\leq\|k_n-k_{\epsilon_n}\|+\epsilon_n\to 0,$$

which leads to the thesis. □

9.2 Constraints and flux conservation

In many cases it is known that the true solution (k,f) lies in some closed and convex set

$$(k,f)\in\Omega_k\times\Omega_f.$$

Therefore, we want to restrict the domain of $J^{\delta,\epsilon}_{\alpha,\beta}$ to $\Omega_k\times\Omega_f$ (for simplicity we assume that $\Omega_k\times\Omega_f\subseteq\mathcal{D}\left(J^{\delta,\epsilon}_{\alpha,\beta}\right)$) and, consequently, our minimization problem becomes a constrained one.

Note that, if $\Omega_k\times\Omega_f$ is compact, the proof of Theorem 9.2 becomes a simple application of the Weierstrass Theorem.

Consider, for example, the framework of convolution such that

$$B(k,f)=k*f. \tag{9.8}$$

Throughout this Section we will assume that k is such that

(i) $k(\mathbf{x})\geq 0\ \forall\mathbf{x}$;

(ii) $\int k(\mathbf{x})\,\mathrm{d}\mathbf{x}=1$.

It is then straightforward to show, using the convolution theorem, the following
Lemma 9.6. *Let k be an integral kernel with compact support and let $g = k * f$, then*

$$\int f = \int g.$$

After a discretization procedure by a collocation method, replacing (9.8) in (9.1), the latter becomes

$$\mathbf{g} = A_k \mathbf{x}.$$

Lemma 9.6 implies that

- A_k has no negative entries;
- If the periodic boundary conditions are imposed, then
  - the row-sum and column-sum of A is 1, i.e., the entries of the vector which discretize k sum up to 1;
  - If $\mathbf{y} = A_k \mathbf{z}$, then $\mathbf{1}^t \mathbf{y} = \mathbf{1}^t \mathbf{z}$, where $\mathbf{1} = (1, 1, \ldots, 1)^t$.

Definition 9.7. *Let $\mathbf{x} \in \mathbb{R}^n$, we call*

$$\textit{flux}\,(\mathbf{x}) = \mathbf{1}^t \mathbf{x}.$$

Remark 9.8. *In the noise-free case and when periodic boundary conditions are employed it holds that*

$$\textit{flux}\,(\mathbf{g}) = \textit{flux}\,(\mathbf{x})\,, \tag{9.9}$$

where $\mathbf{x}, \mathbf{g}$ are the discretization of the true signal f and the noise-free g, respectively.

In the noisy case (9.9) *does not hold in general. Let us call $\mathbf{g}_\delta$ the discretization of g_δ so that*

$$\mathbf{g}_\delta = \mathbf{g} + \boldsymbol{\eta},$$

where $\boldsymbol{\eta}$ represents the discretized noise. It holds

$$\textit{flux}\,(\mathbf{g}_\delta) = \textit{flux}\,(\mathbf{g}) + \textit{flux}\,(\boldsymbol{\eta})\,,$$

but, if we assume that $\boldsymbol{\eta}$ is white Gaussian noise, we have that

$$\textit{flux}\,(\boldsymbol{\eta}) \approx 0.$$

Therefore, in this case we have that

$$\textit{flux}\,(\mathbf{g}_\delta) \approx \textit{flux}\,(\mathbf{x})\,.$$

From Remark 9.8 it follows that we would like to constrain the reconstructed solution to lie in

$$\Omega_F = \left\{ \mathbf{x} \in \mathbb{R}^n \;\middle|\; \textsf{flux}\,(\mathbf{x}) = \textsf{flux}\,(\mathbf{g}_\delta) \right\}. \tag{9.10}$$

Remark 9.9. *The set Ω_F in* (9.10) *is a closed and convex set.*

We now construct P_{Ω_F}, the metric projection over Ω_F. By definition of metric projection we have

$$P_{\Omega_F}(\mathbf{x}) = \arg\min_{\mathbf{y} \in \Omega_F} \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2 .$$

Consider the Fourier matrix $F_1 \in \mathbb{R}^{n \times n}$ defined in (2.6). Let us define $F = \frac{1}{\sqrt{n}} F_1$, F is is a unitary matrix and so $\|F\mathbf{z}\| = \|\mathbf{z}\|$ for all $\mathbf{z}$. Note that the first row of the matrix F is the

constant vector $\frac{1}{\sqrt{n}}\mathbf{1}^t$. Hence the first entry of $F\mathbf{z}$, for some $\mathbf{z} \in \mathbb{R}^n$, is $\frac{1}{\sqrt{n}}\mathbf{1}^t\mathbf{z}$ which is the flux of $\mathbf{z}$ multiplied by $\frac{1}{\sqrt{n}}$, see Definition 9.7. Namely, $\hat{z}_1 = \frac{\mathsf{flux}(\mathbf{z})}{\sqrt{n}}$ for $\hat{\mathbf{z}} = F\mathbf{z}$. This implies that

$$\hat{\Omega}_F = \{\hat{\mathbf{x}} = F\mathbf{x} \mid \mathbf{x} \in \Omega_F\} = \left\{\hat{\mathbf{x}} \in \mathbb{R}^n \mid \hat{x}_1 = \frac{\mathsf{flux}\,(\mathbf{g}_\delta)}{\sqrt{n}}\right\}.$$

Consider now

$$P_{\Omega_F}(\mathbf{x}) = \arg\min_{\mathbf{y}\in\Omega_F} \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2 = \arg\min_{\mathbf{y}\in\Omega_F} \frac{1}{2}\|F\mathbf{x}-F\mathbf{y}\|^2 = F^* \arg\min_{\hat{\mathbf{y}}\in\hat{\Omega}_F} \frac{1}{2}\|\hat{\mathbf{x}}-\hat{\mathbf{y}}\|^2,$$

where we have called $\hat{\mathbf{x}} = F\mathbf{x}, \hat{\mathbf{y}} = F\mathbf{y}$. The solution of the last minimization problem follows straightforward from the definition of $\hat{\Omega}_F$. Defining

$$\hat{\mathbf{z}} = \arg\min_{\hat{\mathbf{y}}\in\hat{\Omega}_F} \frac{1}{2}\|\hat{\mathbf{x}}-\hat{\mathbf{y}}\|^2,$$

the j-th entry of $\hat{\mathbf{z}}$ is

$$\hat{z}_j = \begin{cases} \frac{\mathsf{flux}(\mathbf{g}_\delta)}{\sqrt{n}} & \text{if } j = 1 \\ \hat{x}_j & \text{otherwise.} \end{cases} \tag{9.11}$$

Finally, we have that

$$P_{\Omega_F}(\mathbf{x}) = F^*\hat{\mathbf{z}}, \tag{9.12}$$

with $\hat{\mathbf{z}}$ defined in (9.11).

In practice the computation of P_{Ω_F} does not need any FFT and can be done in $O(n)$ arithmetic operations. Let us call $\mathbf{v}_j$, for $j = 1, \ldots, n$, the vectors of the Fourier basis. The expansion of the vector $\mathbf{x}$ in this base is

$$\mathbf{x} = \sum_{j=1}^{n} \hat{x}_j\mathbf{v}_j = \hat{x}_1\frac{1}{\sqrt{n}}\mathbf{1} + \sum_{j=2}^{n} \hat{x}_j\mathbf{v}_j.$$

According to (9.12) and (9.11), it holds

$$\begin{aligned} P_{\Omega_F}(x) &= \frac{\mathsf{flux}\,(\mathbf{g}_\delta)}{\sqrt{n}}\left(\frac{1}{\sqrt{n}}\mathbf{1}\right) + \sum_{j=2}^{n}\hat{x}_j\mathbf{v}_j, \\ &= \frac{\mathsf{flux}\,(\mathbf{g}_\delta)}{\sqrt{n}}\left(\frac{1}{\sqrt{n}}\mathbf{1}\right) - \hat{x}_1\left(\frac{1}{\sqrt{n}}\mathbf{1}\right) + \hat{x}_1\left(\frac{1}{\sqrt{n}}\mathbf{1}\right) + \sum_{j=2}^{n}\hat{x}_j\mathbf{v}_j \\ &= \frac{(\mathsf{flux}\,(\mathbf{g}_\delta) - \hat{x}_1\sqrt{n})}{n}\mathbf{1} + \mathbf{x} \\ &= \frac{(\mathsf{flux}\,(\mathbf{g}_\delta) - \mathsf{flux}\,(\mathbf{x}))}{n}\mathbf{1} + \mathbf{x}, \end{aligned} \tag{9.13}$$

where the last equation holds recalling that $\hat{x}_1 = \frac{\mathsf{flux}(\mathbf{x})}{\sqrt{n}}$. Note that the computation of $P_{\Omega_F}(\mathbf{x})$ by (9.13) requires only $O(n)$ operations.

9.3 Minimization Algorithm

Computing a minimum of (9.4), especially if the minimization is constrained, can be a difficult problem since it contains both linear and non-linear terms. Moreover, the minimization over two variables can be very challenging. The strategy proposed in [18] was an alternating minimization, where at each step one variable was fixed and the functional was minimized with respect to the other one. With this method it is possible to avoid the complicated minimization over two arguments, but still the minimization over only one variable can be tough. The method we are going to propose decouples the various terms inside (9.4) obtaining a series of simple minimization problems. The main tools to design such decomposition are Augmented Lagrangian and ADMM.

9.3.1 ADMM

We now briefly describe the ADMM algorithm.

First of all we need to define the Augmented Lagrangian related to a constrained minimization problem. Consider

$$\begin{aligned} &x = \arg\min_x f(x), \\ &\text{subject to } Ax = b. \end{aligned} \tag{9.14}$$

The traditional Lagrangian associated with (9.14) is

$$\mathcal{L}(x;\lambda) = f(x) - \langle \lambda, b - Ax \rangle,$$

where λ is the Lagrangian multiplier. The *Augmented Lagrangian* is defined as

$$\mathcal{L}_A(x;\lambda) = f(x) - \langle \lambda, b - Ax \rangle + \frac{\omega}{2} \|b - Ax\|^2,$$

where $\omega > 0$ is called the *penalty parameter*. The Augmented Lagrangian can be seen as the traditional Lagrangian for the problem

$$\begin{aligned} &x = \arg\min_x f(x) + \frac{\omega}{2} \|b - Ax\|^2 \\ &\text{subject to } Ax = b, \end{aligned}$$

which is equivalent to (9.14).

Let us now assume that the problem we have to solve can be written as

$$\begin{aligned} &x = \arg\min_x f(x) + g(z) \\ &\text{subject to } Ax + Bz = c. \end{aligned} \tag{9.15}$$

Firstly, we form the Augmented Lagrangian of (9.15)

$$\mathcal{L}_A(x,z;\lambda) = f(x) + g(z) - \langle \lambda, c - (Ax + Bz) \rangle + \frac{\omega}{2} \|c - (Ax + Bz)\|^2.$$

The ADMM algorithm applied to (9.15) is

Algorithm (ADMM). *Start with initial guesses x_0, z_0, and λ_0 for x, z, and λ, respectively.*

$$
\begin{aligned}
&\texttt{for}\ \ j = 0, 1, 2, \ldots \\
&\quad x^{j+1} = \arg\min_x \mathcal{L}_A\left(x \mid z^j; \lambda^j\right) \\
&\quad z^{j+1} = \arg\min_x \mathcal{L}_A\left(z \mid x^{j+1}; \lambda^j\right) \\
&\quad \lambda^{j+1} = \lambda^j - \omega\left(c - (Ax + Bz)\right) \\
&\texttt{end}
\end{aligned}
$$

Where by $\arg\min_{\bullet} \mathcal{L}_A\left(\bullet|\circ\right)$ *we mean that we minimize the quantity in respect of* $\bullet$ *having fixed the other parameters* $\circ$.

For a recent and comprehensive review on the ADMM method, refer to [21].

It is possible to show that

Theorem 9.10. *With the same notation as above, if f and g are closed, proper, and convex functions and if the unaugmented Lagrangian has a saddle point, then the ADMM method converges to a solution of* (9.15).

Constrained Minimization We now discuss how to use the ADMM algorithm for solving a constrained minimization problem. Suppose that we want to constrain a minimization problem like (9.14), so that our minimizer lies in some closed and convex set $\Omega \subset \mathcal{D}(f)$

$$
\begin{aligned}
x &= \arg\min_{x \in \Omega} f(x), \\
&\text{subject to } Ax = b.
\end{aligned}
\tag{9.16}
$$

Let us write (9.16) in an equivalent way

$$
\begin{aligned}
(x, \tilde{x}) &= \arg\min_{\tilde{x} \in \Omega, x} f(x), \\
&\text{subject to } Ax = b \text{ and } \tilde{x} = x.
\end{aligned}
\tag{9.17}
$$

The Augmented Lagrangian associated with (9.17) is

$$
\mathcal{L}_A(\tilde{x}, x; \lambda, \xi) = f(x) - \langle \lambda, \tilde{x} - x \rangle + \frac{\omega_1}{2} \|\tilde{x} - x\|^2 - \langle \xi, b - Ax \rangle + \frac{\omega_2}{2} \|b - Ax\|^2 .
$$

The ADMM applied to (9.17) leads to the following algorithm:

Algorithm. *Let x_0, λ_0, and ξ_0 be initial guesses for x, λ, and ξ, respectively*

$$
\begin{aligned}
&\texttt{for}\ \ j = 0, 1, \ldots \\
&\quad \tilde{x}_{j+1} = \arg\min_{\tilde{x} \in \Omega} \mathcal{L}_A\left(\tilde{x} \mid x_j; \lambda_j, \xi_j\right) \\
&\quad x_{j+1} = \arg\min_x \mathcal{L}_A\left(x \mid \tilde{x}_{j+1}; \lambda_j, \xi_j\right) \\
&\quad \lambda_{j+1} = \lambda_j - \omega_1\left(\tilde{x}_{j+1} - x_{j+1}\right) \\
&\quad \xi_{j+1} = \xi_j - \omega_2\left(b - Ax_{j+1}\right) \\
&\texttt{end}
\end{aligned}
$$

It is easy to show that $\tilde{x}_{j+1}$ is obtained by

$$
\tilde{x}_{j+1} = P_\Omega\left(x_j + \frac{\lambda_j}{\omega_1}\right).
$$

In this way we are able to easily deal with the constrained optimization problem.

This approach is feasible if the projection P_Ω is easily performed. On the other hand if the projection into Ω is too complicated the algorithm above might not be attractive. Nevertheless, whenever Ω can be written as the intersection of two or more closed and convex sets, i.e.,

$$\Omega = \bigcap_{l=1}^{L} \Omega^{(l)},$$

if $P_{\Omega^{(l)}}$ is easily performed, then we can still use ADMM to solve the constrained minimization problem.

For the sake of simplicity we fix $L = 2$. Consider the minimization problem

$$\begin{aligned} &\min_x f(x) \\ &\text{s.t. } x \in \Omega^{(1)} \cap \Omega^{(2)}, \end{aligned}$$

which is equivalent to the following

$$\begin{aligned} &\min_{x, x^{(1)}, x^{(2)}} f(x) \\ &\text{s.t. } x^{(1)} \in \Omega^{(1)}, x^{(2)} \in \Omega^{(2)}, x = x^{(1)}, x = x^{(2)}. \end{aligned} \tag{9.18}$$

In this way we have separated the two constraints on $\Omega^{(1)}$ and $\Omega^{(2)}$; hopefully the projection on each set is easier to compute than the projection over the intersection.

The Augmented Lagrangian of the new minimization problem (9.18) is

$$\mathcal{L}_A\left(x, x^{(1)}, x^{(2)}; \lambda, \theta\right) = f(x) - \left\langle \lambda, x^{(1)} - x \right\rangle + \frac{\omega_1}{2}\left\| x^{(1)} - x \right\|^2 - \left\langle \theta, x^{(2)} - x \right\rangle + \frac{\omega_2}{2}\left\| x^{(2)} - x \right\|^2.$$

We can now write the ADMM iterations for this $\mathcal{L}_A$.

Algorithm 9.1. *Given f_0, λ_0, and θ_0 initial guesses for f, λ, and θ, respectively. Let $\omega_1, \omega_2 > 0$ be real constant numbers.*

`for` $j = 0, 1, \ldots$

$$\begin{pmatrix} x_{j+1}^{(1)} \\ x_{j+1}^{(2)} \end{pmatrix} = \arg\min_{x^{(1)}, x^{(2)}} \mathcal{L}_A\left(x^{(1)}, x^{(2)} \middle| x_j; \lambda_j, \theta_j\right)$$

$$x_{j+1} = \arg\min_x \mathcal{L}_A\left(x \middle| x_{j+1}^{(1)}, x_{j+1}^{(2)}; \lambda_j, \theta_j\right)$$

$$\lambda_{j+1} = \lambda_j - \omega_1\left(x_{j+1}^{(1)} - x_{j+1}\right)$$

$$\theta_{j+1} = \theta_j - \omega_2\left(x_{j+1}^{(2)} - x_{j+1}\right)$$

`end`

The first minimization problem decouples and the solutions are simply obtained by

$$x_{j+1}^{(1)} = P_{\Omega^{(1)}}\left(x_j + \frac{\lambda_j}{\omega_1}\right), \qquad x_{j+1}^{(2)} = P_{\Omega^{(2)}}\left(x_j + \frac{\theta_j}{\omega_2}\right).$$

We do not consider the minimization in respect to x since it is not relevant for our purpose.

9.3.2 The proposed Algorithm

We now consider the finite dimensional case, in particular for simplicity we assume $\mathbf{f}, \mathbf{k} \in \mathbb{R}^{n\times n}$ (the extension to the case where $\mathbf{f}$ and $\mathbf{k}$ belong to different spaces and are not square is straightforward). We apply the ADMM to the Augmented Lagrangian related to the following constrained minimization problem

$$(\mathbf{k}^*, \mathbf{f}^*) = \arg \min_{\mathbf{k}\in\Omega_{\mathbf{k}}, \mathbf{f}\in\Omega_{\mathbf{f}}} J^{\delta,\epsilon}_{\alpha,\beta}(\mathbf{k}, \mathbf{f}), \tag{9.19}$$

where, with abuse of notation, we call $J^{\delta,\epsilon}_{\alpha,\beta}$ the discretization of the function defined in (9.4) and $\Omega_{\mathbf{k}}, \Omega_{\mathbf{f}} \subset \mathbb{R}^{n\times n}$ are closed and convex sets.

Please note that Theorem 9.10 does not assure the convergence of the ADMM in this case since $J^{\delta,\epsilon}_{\alpha,\beta}$ is non-convex, so we are going to need a different result for the convergence. In particular we are going to need a further assumption on $\Omega_{\mathbf{k}} \times \Omega_{\mathbf{f}}$.

Before covering the convergence property of the proposed algorithm we explicitly formulate all its ingredients.

The *isotropic Total Variation* operator in this space is defined as follows. Let $\mathbf{x} \in \mathbb{R}^{n\times n}$

$$TV(\mathbf{x}) = \sum_{i=1}^{n^2} \|D_i\mathbf{x}\|,$$

where $D_i\mathbf{x} = \left(\left(D^{(1)}x\right)_i, \left(D^{(2)}\mathbf{x}\right)_i\right)^t \in \mathbb{R}^2$ is

$$\begin{aligned} \left(D^{(1)}\mathbf{x}\right)_i &= \begin{cases} x_{i+n} - x_i, & \text{if } 1 \le i \le n(n-1) \\ x_{\mathrm{mod(i,n)}} - x_i, & \text{otherwise} \end{cases} \\ \left(D^{(2)}\mathbf{x}\right)_i &= \begin{cases} x_{i+1} - x_i, & \text{if } \mathrm{mod}(i,n) \neq 0 \\ x_{i-n+1} - x_i, & \text{otherwise.} \end{cases} \end{aligned} \tag{9.20}$$

With abuse of notation we will write

$$\|\mathbf{x}\|_{TV} = TV(\mathbf{x}),$$

also in the finite dimensional case.

Let us rewrite (9.19) in an equivalent way

$$\begin{aligned} (\mathbf{k}^*, \mathbf{f}^*) &= \arg \min_{\mathbf{k}\in\Omega_{\mathbf{k}}, \mathbf{f}\in\Omega_{\mathbf{f}}} \|B(\mathbf{k}, \mathbf{f}) - \mathbf{g}_\delta\|^2 + \alpha\left(\|\mathbf{f}\|^2 + \|\mathbf{f}\|_{TV}\right) + \gamma\|\mathbf{k} - \mathbf{k}_\epsilon\|^2 + \beta\|\mathbf{k}\|_{TV} \\ &= \arg \min_{\substack{\tilde{\mathbf{k}}\in\Omega_{\mathbf{k}}, \hat{\mathbf{f}}\in\Omega_{\mathbf{f}} \\ \hat{\mathbf{k}}, \hat{\mathbf{f}}, \mathbf{k}, \mathbf{f}}} \left\{\|B(\mathbf{k}, \mathbf{f}) - \mathbf{g}_\delta\|^2 + \alpha\left(\|\mathbf{f}\|^2 + \left\|\hat{\mathbf{f}}\right\|_{TV}\right) + \gamma\|\mathbf{k} - \mathbf{k}_\epsilon\|^2 + \beta\left\|\hat{\mathbf{k}}\right\|_{TV}, \right. \\ &\qquad \left. \mathbf{k} = \tilde{\mathbf{k}}, \mathbf{f} = \tilde{\mathbf{f}}, \mathbf{k} = \hat{\mathbf{k}}, \mathbf{f} = \hat{\mathbf{f}}\right\}, \end{aligned}$$

where $N = n^2$, $\mathbf{f}, \tilde{\mathbf{f}}, \hat{\mathbf{f}}, \mathbf{k}, \tilde{\mathbf{k}}, \hat{\mathbf{k}} \in \mathbb{R}^N$.

We now write the Augmented Lagrangian of the minimization above

$$\begin{aligned}
&\mathcal{L}_A\left(\tilde{\mathbf{f}},\hat{\mathbf{f}},\mathbf{f},\tilde{\mathbf{k}},\hat{\mathbf{k}},\mathbf{k};\boldsymbol{\lambda},\boldsymbol{\xi},\boldsymbol{\zeta},\boldsymbol{\mu}\right)\\
&=\|B\left(\mathbf{k},\mathbf{f}\right)-\mathbf{g}_\delta\|^2+\alpha\left(\|\mathbf{f}\|^2+\left\|\hat{\mathbf{f}}\right\|_{TV}\right)+\gamma\left\|\mathbf{k}-\mathbf{k}_\epsilon\right\|+\beta\left\|\hat{\mathbf{k}}\right\|_{TV}\\
&+\frac{\omega_1}{2}\left\|\tilde{\mathbf{f}}-\mathbf{f}\right\|^2-\left\langle\boldsymbol{\lambda},\tilde{\mathbf{f}}-\mathbf{f}\right\rangle+\frac{\omega_2}{2}\left\|\hat{\mathbf{f}}-\mathbf{f}\right\|^2-\left\langle\boldsymbol{\xi},\hat{\mathbf{f}}-\mathbf{f}\right\rangle\\
&+\frac{\omega_3}{2}\left\|\tilde{\mathbf{k}}-\mathbf{k}\right\|^2-\left\langle\boldsymbol{\zeta},\tilde{\mathbf{k}}-\mathbf{k}\right\rangle+\frac{\omega_4}{2}\left\|\hat{\mathbf{k}}-\mathbf{k}\right\|^2-\left\langle\boldsymbol{\mu},\hat{\mathbf{k}}-\mathbf{k}\right\rangle,
\end{aligned}$$

where $\boldsymbol{\lambda},\boldsymbol{\zeta},\boldsymbol{\xi},\boldsymbol{\mu}\in\mathbb{R}^N$.

We can apply the ADMM algorithm obtaining

Algorithm 9.2 (SeB-A). *Given* $\mathbf{f}_0$, $\mathbf{k}_0$, $\boldsymbol{\lambda}_0$, $\boldsymbol{\xi}_0$, $\boldsymbol{\zeta}_0$, *and* $\boldsymbol{\mu}_0$ *initial guesses for* $\mathbf{f}$, $\mathbf{k}$, $\boldsymbol{\lambda}$, $\boldsymbol{\xi}$, $\boldsymbol{\zeta}$, *and* $\boldsymbol{\mu}$, *respectively. Let* $\omega_1,\omega_2,\omega_3,\omega_4>0$ *be real fixed numbers.*

for $j=0,1,\ldots$

$$\begin{aligned}
&\begin{pmatrix}\tilde{\mathbf{f}}_{j+1}\\ \hat{\mathbf{f}}_{j+1}\\ \mathbf{k}_{j+1}\end{pmatrix}=\arg\min_{\tilde{\mathbf{f}},\hat{\mathbf{f}},\mathbf{k}}\mathcal{L}_A\left(\tilde{\mathbf{f}},\hat{\mathbf{f}},\mathbf{k}\middle|\tilde{\mathbf{k}}_j,\hat{\mathbf{k}}_j,\mathbf{f}_j;\boldsymbol{\lambda}_j,\boldsymbol{\xi}_j,\boldsymbol{\zeta}_j,\boldsymbol{\mu}_j\right)\\
&\begin{pmatrix}\tilde{\mathbf{k}}_{j+1}\\ \hat{\mathbf{k}}_{j+1}\\ \mathbf{f}_{j+1}\end{pmatrix}=\arg\min_{\tilde{\mathbf{k}},\hat{\mathbf{k}},\mathbf{f}}\mathcal{L}_A\left(\tilde{\mathbf{k}},\hat{\mathbf{k}},\mathbf{f}\middle|\tilde{\mathbf{f}}_{j+1},\hat{\mathbf{f}}_{j+1},\mathbf{k}_{j+1};\boldsymbol{\lambda}_j,\boldsymbol{\xi}_j,\boldsymbol{\zeta}_j,\boldsymbol{\mu}_j\right)\\
&\boldsymbol{\lambda}_{j+1}=\boldsymbol{\lambda}_j-\omega_1\left(\tilde{\mathbf{f}}_{j+1}-\mathbf{f}_{j+1}\right)\\
&\boldsymbol{\xi}_{j+1}=\xi_j-\omega_2\left(\hat{\mathbf{f}}_{j+1}-\mathbf{f}_{j+1}\right)\\
&\boldsymbol{\zeta}_{j+1}=\boldsymbol{\zeta}_j-\omega_3\left(\tilde{\mathbf{k}}_{j+1}-\mathbf{k}_{j+1}\right)\\
&\boldsymbol{\mu}_{j+1}=\boldsymbol{\mu}_j-\omega_4\left(\hat{\mathbf{k}}_{j+1}-\mathbf{k}_{j+1}\right)
\end{aligned}$$

end

We call this method SeB-A as for *Semi-blind ADMM*. We now formulate some assumptions that we are going to need in the following.

Assumption 9.2. $B\left(\mathbf{k},\mathbf{f}\right)$ *is bilinear.*

Remark 9.11. *The semi-blind deconvolution problem we are interested in satisfies Assumption 9.2.*

Under Assumption 9.2 most of the minimization above are easily computed. We have that

$$\begin{aligned}
\tilde{\mathbf{f}}_{j+1}&=P_{\Omega_{\mathbf{f}}}\left(\mathbf{f}_j+\frac{\boldsymbol{\lambda}_j}{\omega_1}\right)\\
\mathbf{k}_{j+1}&=\left(2A^*_{\mathbf{f}_j}A_{\mathbf{f}_j}+2\gamma I+\left(\omega_3+\omega_4\right)I\right)^{-1}\left(2A^*_{\mathbf{f}_j}\mathbf{g}_\delta+2\gamma\mathbf{k}_\epsilon-\boldsymbol{\zeta}_j\right.\\
&\qquad\left.+\omega_3\tilde{\mathbf{k}}_j-\boldsymbol{\mu}_j+\omega_4\hat{\mathbf{k}}_j\right)\\
\tilde{\mathbf{k}}_{j+1}&=P_{\Omega_{\mathbf{k}}}\left(\mathbf{k}_{j+1}+\frac{\boldsymbol{\zeta}_j}{\omega_3}\right)\\
\mathbf{f}_{j+1}&=\left(2A^*_{\mathbf{k}_{j+1}}A_{\mathbf{k}_{j+1}}+2\alpha I+\left(\omega_1+\omega_2\right)I\right)^{-1}\left(2A^*_{\mathbf{k}_{j+1}}\mathbf{g}_\delta-\boldsymbol{\lambda}\right.\\
&\qquad\left.+\omega_1\tilde{\mathbf{f}}_{j+1}-\boldsymbol{\xi}_j+\omega_2\hat{\mathbf{f}}_{j+1}\right)
\end{aligned}$$

Where by $A_\bullet$ we indicate the linear operator obtained from $B\left(\mathbf{k},\mathbf{f}\right)$ by fixing $\bullet$.

On the other hand the minimizations with respect of $\hat{\mathbf{f}}$ and $\hat{\mathbf{k}}$ are non trivial, in fact

$$\begin{aligned}
\hat{\mathbf{f}}_{j+1} &= \arg\min_{\hat{\mathbf{f}}} \alpha \left\|\hat{\mathbf{f}}\right\|_{TV} + \frac{\omega_2}{2}\left\|\hat{\mathbf{f}} - \mathbf{f}_j\right\|^2 - \left\langle \boldsymbol{\xi}_j, \hat{\mathbf{f}} - \mathbf{f}_j\right\rangle \\
&= \arg\min_{\hat{\mathbf{f}}} \left\|\hat{\mathbf{f}}\right\|_{TV} + \frac{\omega_2}{2\alpha}\left\|\hat{\mathbf{f}} - \left(\mathbf{f}_j + \frac{\boldsymbol{\xi}_j}{\omega_2}\right)\right\|^2 \\
\hat{\mathbf{k}}_{j+1} &= \arg\min_{\hat{\mathbf{k}}} \beta \left\|\hat{\mathbf{k}}\right\|_{TV} + \frac{\omega_4}{2}\left\|\hat{\mathbf{k}} - \mathbf{k}_{j+1}\right\|^2 - \left\langle \boldsymbol{\mu}_j, \hat{\mathbf{k}} - \mathbf{k}_{j+1}\right\rangle \\
&= \arg\min_{\hat{\mathbf{k}}} \left\|\hat{\mathbf{k}}\right\|_{TV} + \frac{\omega_4}{2\beta}\left\|\hat{\mathbf{k}} - \left(\mathbf{k}_{j+1} + \frac{\boldsymbol{\mu}_j}{\omega_4}\right)\right\|^2
\end{aligned}$$

To solve this minimization problems we can use, e.g., the ADMM algorithm. Since, however, in this case the functional are proper and convex the convergence is assured by the classical ADMM theory, see, e.g., [21]. For completeness we describe here this approach.

Consider the minimization problem

$$\mathbf{x} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_{TV} + c\|\mathbf{x} - \mathbf{y}\|^2, \tag{9.21}$$

where $c > 0$ is a constant. We can rewrite (9.21) as

$$\mathbf{x} = \arg\min_{\mathbf{x},\hat{\mathbf{x}}} \left\{ \sum_{i=1}^{N} \|(\hat{\mathbf{x}})_i\| + c\|\mathbf{x} - \mathbf{y}\|^2, \ (\hat{\mathbf{x}})_i = D_i\mathbf{x} \right\}.$$

The related augmented Lagrangian is

$$\mathcal{L}_A(\mathbf{x}, \hat{\mathbf{x}}; \boldsymbol{\lambda}) = \sum_{i=1}^{N} \|(\hat{\mathbf{x}})_i\| + c\|\mathbf{x} - \mathbf{y}\|^2 + \sum_{i=1}^{N} \left(\frac{\omega}{2}\|\hat{\mathbf{x}}_i - D_i\mathbf{x}\|^2 - \langle (\boldsymbol{\lambda})_i, \hat{\mathbf{x}}_i - D_i\mathbf{x}\rangle \right).$$

The resulting ADMM algorithm is the following
Algorithm (ADMM for TV optimization)**.** *Given* $\mathbf{x}_0$ *and* $\boldsymbol{\lambda}_0$ *initial guesses for* $\mathbf{x}$ *and* λ*, respectively. Let* $\omega > 0$ *be a real fixed number.*

for $j = 0, 1, \ldots$
$\hat{\mathbf{x}}_{j+1} = \arg\min_{\hat{\mathbf{x}}} \sum_{i=1}^{N} \left(\|\hat{\mathbf{x}}_i\| + \frac{\omega}{2}\|\hat{\mathbf{x}}_i - D_i\mathbf{x}_j\|^2 - \langle (\boldsymbol{\lambda}_j)_i, \hat{\mathbf{x}}_i - D_i\mathbf{x}_j\rangle \right)$
$\mathbf{x}_{j+1} = \arg\min_{\mathbf{x}} c\|\mathbf{x} - \mathbf{y}\|^2 + \sum_{i=1}^{N} \left(\frac{\omega}{2}\|(\hat{\mathbf{x}}_{j+1})_i - D_i\mathbf{x}\|^2 - \langle (\boldsymbol{\lambda}_j)_i, (\hat{\mathbf{x}}_{j+1})_i - D_i\mathbf{x}\rangle \right)$
$\boldsymbol{\lambda}_{j+1} = \boldsymbol{\lambda}_j - \omega(\hat{\mathbf{x}}_{j+1} - D\mathbf{x}_{j+1})$
end

The minimization above are easily computed. The minimization with respect to $\hat{\mathbf{x}}$ decouples in N subproblems which are easily solved using a two-dimensional shrinkage and the minimization with respect to $\mathbf{x}$ can be achieved by solving a linear system. In particular

$$\begin{aligned}
(\hat{\mathbf{x}}_{j+1})_i &= \frac{D_i\mathbf{x}_j + \frac{(\boldsymbol{\lambda}_j)_i}{\omega}}{\left\|D_i\mathbf{x}_j + \frac{(\boldsymbol{\lambda}_j)_i}{\omega}\right\|} \circ \left(\left\|D_i\mathbf{x}_j + \frac{(\boldsymbol{\lambda}_j)_i}{\omega}\right\| - \frac{1}{c\omega} \right)_+ \\
\mathbf{x}_{j+1} &= (2cI + \omega D^*D)^{-1}(2c\mathbf{y} + \omega D^*\hat{\mathbf{x}}_{j+1} - D^*\boldsymbol{\lambda}_j)
\end{aligned}$$

In this setup D is the linear operator which maps $\mathbb{R}^N$ into $\mathbb{R}^{N\times 2}$ defined as $\begin{pmatrix} D^{(1)} \\ D^{(2)} \end{pmatrix}$.

We now prove the convergence of SeB-A Algorithm 9.2. This proof is very technical and inspired by [87].

Remark 9.12. *The convergence of Algorithm 9.2 is proven only under Conjecture 9.13, i.e., that the norms of the iterates* $\mathbf{f}_j$ *and* $\mathbf{k}_j$ *remain bounded. While this seems a very strong requirement, from the numerical experiments we can see that this condition is always satisfied. We provide some simple bounds* φ_k *and* φ_f *for the norms of* $\mathbf{k}_j$ *and* $\mathbf{f}_j$*, respectively. In Section 9.4 we show that this bounds are far from being violated in all the computed examples.*

Proof of convergence

We now give prove that Algorithm 9.2 converges to a stationary point of $J^{\delta,\epsilon}_{\alpha,\beta}(\mathbf{k},\mathbf{f})$.

We first analyze the unconstrained case, i.e., we consider the minimization problem

$$(\mathbf{k}^*,\mathbf{f}^*) = \arg\min_{\mathbf{k},\mathbf{f}} \|B(\mathbf{k},\mathbf{f}) - \mathbf{g}_\delta\|^2 + \alpha\left(\|\mathbf{f}\|^2 + \|\mathbf{f}\|_{TV}\right) + \gamma\|\mathbf{k}-\mathbf{k}_\epsilon\|^2 + \beta\|\mathbf{k}\|_{TV},$$

Let us rewrite the minimization problem above in an equivalent way

$$\begin{aligned}(\mathbf{k}^*,\mathbf{f}^*) = \arg\min_{\substack{\mathbf{k},\mathbf{f}\\ \hat{\mathbf{k}},\hat{\mathbf{f}}}} \Big\{ &\|B(\mathbf{k},\mathbf{f}) - \mathbf{g}_\delta\|^2 + \alpha\left(\|\mathbf{f}\|^2 + \left\|\hat{\mathbf{f}}\right\|_{TV}\right)\\ &+\gamma\|\mathbf{k}-\mathbf{k}_\epsilon\|^2 + \beta\left\|\hat{\mathbf{k}}\right\|_{TV},\ \hat{\mathbf{f}}=\mathbf{f}, \hat{\mathbf{k}}=\mathbf{k}\Big\}.\end{aligned} \tag{9.22}$$

We now form the augmented Lagrangian related to the minimization problem (9.22), where, without loss of generality, we have chosen the same ω for all the augmentation terms

$$\begin{aligned}\mathcal{L}_A\left(\mathbf{f},\mathbf{k},\hat{\mathbf{f}},\hat{\mathbf{k}};\boldsymbol{\xi},\boldsymbol{\mu}\right) = &\|B(\mathbf{k},\mathbf{f}) - \mathbf{g}_\delta\|^2 + \alpha\left(\|\mathbf{f}\|^2 + \left\|\hat{\mathbf{f}}\right\|_{TV}\right)\\ &+\gamma\|\mathbf{k}-\mathbf{k}_\epsilon\|^2 + \beta\left\|\hat{\mathbf{k}}\right\|_{TV}\\ &+\frac{\omega}{2}\left\|\hat{\mathbf{f}}-\mathbf{f}\right\|^2 - \left\langle\boldsymbol{\xi},\hat{\mathbf{f}}-\mathbf{f}\right\rangle\\ &+\frac{\omega}{2}\left\|\hat{\mathbf{k}}-\mathbf{k}\right\|^2 - \left\langle\boldsymbol{\mu},\hat{\mathbf{k}}-\mathbf{k}\right\rangle.\end{aligned} \tag{9.23}$$

Thus the unconstrained algorithm becomes

Algorithm 9.3. *Given* $\mathbf{f}_0$, $\mathbf{k}_0$, $\boldsymbol{\xi}_0$, *and* $\boldsymbol{\mu}_0$ *initial guesses for* $\mathbf{f}$, $\mathbf{k}$, $\boldsymbol{\xi}$, *and* $\boldsymbol{\mu}$, *respectively.*

for $j = 0, 1, \ldots$

$$\begin{pmatrix}\hat{\mathbf{f}}_{j+1}\\ \mathbf{k}_{j+1}\end{pmatrix} = \arg\min_{\hat{\mathbf{f}},\mathbf{k}} \mathcal{L}_A\left(\hat{\mathbf{f}},\mathbf{k}\middle|\hat{\mathbf{k}}_j,\mathbf{f}_j;\boldsymbol{\xi}_j,\boldsymbol{\mu}_j\right)$$

$$\begin{pmatrix}\hat{\mathbf{k}}_{j+1}\\ \mathbf{f}_{j+1}\end{pmatrix} = \arg\min_{\hat{\mathbf{k}},\mathbf{f}} \mathcal{L}_A\left(\hat{\mathbf{k}},\mathbf{f}\middle|\hat{\mathbf{f}}_{j+1},\mathbf{k}_{j+1};\boldsymbol{\xi}_j,\boldsymbol{\mu}_j\right)$$

$$\boldsymbol{\xi}_{j+1} = \boldsymbol{\xi}_j - \omega\left(\hat{\mathbf{f}}_{j+1} - \mathbf{f}_{j+1}\right)$$

$$\boldsymbol{\mu}_{j+1} = \boldsymbol{\mu}_j - \omega\left(\hat{\mathbf{k}}_{j+1} - \mathbf{k}_{j+1}\right)$$

end

This simplified version of our method is an algorithm to compute a solution of the unconstrained minimization problem

$$\arg\min_{\mathbf{k},\mathbf{f}} J^{\delta,\epsilon}_{\alpha,\beta}(\mathbf{k},\mathbf{f}).$$

In our proofs we are not going to consider the constrain $(\mathbf{k},\mathbf{f}) \in \Omega_{\mathbf{k}} \times \Omega_{\mathbf{f}}$. We are going to insert this constraint only at the very end.

To proceed we need some further assumptions
Assumption 9.3. *We assume that*

(i) If $\mathbf{k} = \mathbf{0}$ *or* $\mathbf{f} = \mathbf{0}$ *then* $B(\mathbf{k},\mathbf{f}) = \mathbf{0}$*;*

(ii) If for a certain set $K = \{\mathbf{k}^{(l)}, l = 1,2,\ldots\}$ *it holds that* $\left\|\mathbf{k}^{(l)}\right\| < C_K$*, where* C_K *is a constant, then the operators* $A_{\mathbf{k}^{(l)}} = B\left(\mathbf{k}^{(l)},\cdot\right)$*, which are linear in force of Assumption 9.2, have bounded norm, i.e. there exists a constant* C *such that* $\forall \mathbf{f}$ *and* $\forall \mathbf{k} \in K$ *it holds* $\|B(\mathbf{k},\mathbf{f})\| < C\|\mathbf{f}\|$*.*

Similarly assume that for a certain set $F = \{\mathbf{f}^{(l)}, l = 1,2,\ldots\}$ *it holds that* $\left\|\mathbf{f}^{(l)}\right\| < C_F$*, where* C_F *is a constant, then the operators* $A_{\mathbf{f}^{(l)}} = B\left(\cdot,\mathbf{f}^{(l)}\right)$*, which are linear in force of Assumption 9.2, have bounded norm, i.e. there exists a constant* C *such that* $\forall \mathbf{k}$ *and* $\forall \mathbf{f} \in F$ *it holds* $\|B(\mathbf{k},\mathbf{f})\| < C\|\mathbf{k}\|$*;*

(iii) The parameter ω *is big enough so that*

$$\|B(\mathbf{k},\mathbf{f}) - \mathbf{g}_\delta\|^2 + \alpha\|\mathbf{f}\|^2 + \frac{\omega}{2}\left\|\hat{\mathbf{f}} - \mathbf{f}\right\|^2 - \left\langle \boldsymbol{\xi}, \hat{\mathbf{f}} - \mathbf{f}\right\rangle,$$

$$\|B(\mathbf{k},\mathbf{f}) - \mathbf{g}_\delta\|^2 + \gamma\|\mathbf{k} - \mathbf{k}_\epsilon\|^2 + \frac{\omega}{2}\left\|\hat{\mathbf{k}} - \mathbf{k}\right\|^2 - \left\langle \boldsymbol{\mu}, \hat{\mathbf{k}} - \mathbf{k}\right\rangle$$

are strongly convex with respect to $\mathbf{f}$ *and* $\mathbf{k}$*, respectively with modulus* ρ*.*

We are going now to conjecture that the norms of the iterates $\mathbf{f}_j$ and $\mathbf{k}_j$ are bounded. Moreover, we also derive a bound in the case in which the flux constraint (9.9) is imposed.
Conjecture 9.13. *The norm of the iterates* $\mathbf{f}_j$ *and* $\mathbf{k}_j$ *generated by Algorithm 9.3 are bounded. Moreover, if the flux and nonnegativity constraints are imposed, then the flux bounds the norm of the iterates. In particular, if we constrain the flux of* $\mathbf{k}$ *to be* $\varphi_{\mathbf{k}}$ *then*

$$\|\mathbf{k}_j\| \leq \varphi_{\mathbf{k}} \qquad \forall\, j.$$

If we constrain the flux of $\mathbf{f}$ *to be* $\varphi_{\mathbf{f}}$ *then*

$$\|\mathbf{f}_j\| \leq \varphi_{\mathbf{f}} \qquad \forall\, j.$$

The bounds proposed above are derived by the following argument. We consider $\mathbf{f}$, but the extension to $\mathbf{k}$ is trivial. Recalling that $\forall \mathbf{z} \in \mathbb{R}^N$ it holds that $\|\mathbf{z}\| \leq \|\mathbf{z}\|_1$. Since we are imposing that $\mathsf{flux}(\mathbf{f}) = \varphi_{\mathbf{f}}$ and that $\mathbf{f} \geq 0$, it yields

$$\|\mathbf{f}\| \leq \|\mathbf{f}\|_1 = \mathsf{flux}(\mathbf{f}) = \varphi_{\mathbf{f}}.$$

Conjecture 9.13 seems indeed strong, however, as shown in Section 9.4, it is always satisfied in our numerical examples.

For convenience we define

$$\phi(\mathbf{f},\mathbf{k}) = \|B(\mathbf{k},\mathbf{f}) - \mathbf{g}_\delta\|^2 + \alpha\|\mathbf{f}\|^2 + \gamma\|\mathbf{k}-\mathbf{k}_\epsilon\|^2, \tag{9.24}$$

$$\psi_\alpha(\mathbf{f}) = \alpha\|\mathbf{f}\|_{TV}, \tag{9.25}$$

$$\psi_\beta(\mathbf{k}) = \beta\|\mathbf{k}\|_{TV}. \tag{9.26}$$

We now prove an auxiliary result which we need for the following.

Lemma 9.14. *Let $\boldsymbol{\xi}_j, \boldsymbol{\mu}_j, \mathbf{f}_j, \mathbf{k}_j$ be the iterations generated by Algorithm 9.3. Assume that Assumptions 9.1-9.3 hold and assume that Conjecture 9.13 holds. Then we have*

$$\left\|\boldsymbol{\xi}_{j+1} - \boldsymbol{\xi}_j\right\| \le C\|\mathbf{f}_{j+1} - \mathbf{f}_j\|,$$

$$\left\|\boldsymbol{\mu}_{j+1} - \boldsymbol{\mu}_j\right\| \le C\left\|\hat{\mathbf{k}}_{j+1} - \hat{\mathbf{k}}_j\right\|$$

where $C > 0$ is a constant.

Proof. We prove the first inequality.

Consider the optimality condition for $\mathbf{f}_{j+1}$ obtained differentiating (9.23)

$$\mathbf{0} = \nabla_{\mathbf{f}}\phi(\mathbf{f}_{j+1},\mathbf{k}_{j+1}) + \boldsymbol{\xi}_j - \omega\left(\hat{\mathbf{f}}_{j+1} - \mathbf{f}_{j+1}\right),$$

where ϕ is defined in (9.24). Using the update rule for $\boldsymbol{\xi}_{j+1}$ we get

$$-\boldsymbol{\xi}_{j+1} = \nabla_{\mathbf{f}}\phi(\mathbf{f}_{j+1},\mathbf{k}_{j+1}). \tag{9.27}$$

Combining Conjecture 9.13 with Assumption 9.3(ii), we get that the linear operators $\{\nabla_{\mathbf{f}}\phi(\cdot,\mathbf{k}_j)\}_j$ have uniformly bounded norm, i.e., there exists a constant $C_{\mathbf{f}} > 0$ such that

$$\|\nabla_{\mathbf{f}}\phi(\mathbf{x},\mathbf{k}_{j+1}) - \nabla_{\mathbf{f}}\phi(\mathbf{y},\mathbf{k}_{j+1})\| \le C_{\mathbf{f}}\|\mathbf{x}-\mathbf{y}\|.$$

Hence we have

$$\left\|\boldsymbol{\xi}_{j+1} - \boldsymbol{\xi}_j\right\| = \|\nabla_{\mathbf{f}}\phi(\mathbf{f}_{j+1},\mathbf{k}_{j+1}) - \nabla_{\mathbf{f}}\phi(\mathbf{f}_j,\mathbf{k}_{j+1})\| \le C_{\mathbf{f}}\|\mathbf{f}_{j+1} - \mathbf{f}_j\|.$$

We now move to the second inequality.

Considering the optimality condition of (9.23) for $\hat{\mathbf{k}}_{j+1}$ and denoting with $\partial\psi_\beta$ the subdifferential of ψ_β defined in (9.26), we get

$$\begin{aligned}\mathbf{0} &\in \partial\psi_\beta\left(\hat{\mathbf{k}}_{j+1}\right) - \boldsymbol{\mu}_j + \omega\left(\hat{\mathbf{k}}_{j+1} - \mathbf{k}_{j+1}\right)\\ &= \partial\psi_\beta\left(\hat{\mathbf{k}}_{j+1}\right) - \boldsymbol{\mu}_{j+1},\end{aligned}$$

in other words

$$\boldsymbol{\mu}_{j+1} \in \partial\psi_\beta\left(\hat{\mathbf{k}}_{j+1}\right).$$

Thus it holds

$$\boldsymbol{\mu}_{j+1} - \boldsymbol{\mu}_j \in \partial\psi_\beta\left(\hat{\mathbf{k}}_{j+1}\right) - \partial\psi_\beta\left(\hat{\mathbf{k}}_j\right),$$

Hence

$$\left\|\boldsymbol{\mu}_{j+1} - \boldsymbol{\mu}_j\right\| \le \sup\left\|\partial\psi_\beta\left(\hat{\mathbf{k}}_{j+1}\right) - \partial\psi_\beta\left(\hat{\mathbf{k}}_j\right)\right\|.$$

By Conjecture 9.13 we have that $\left\|\hat{\mathbf{k}}_j\right\|$ is uniformly bounded and thus there exists $C_{\hat{\mathbf{k}}} > 0$ such that

$$\left\|\boldsymbol{\mu}_{j+1} - \boldsymbol{\mu}_j\right\| \leq C_{\hat{\mathbf{k}}} \left\|\hat{\mathbf{k}}_{j+1} - \hat{\mathbf{k}}_j\right\|.$$

Calling $C = \max\left\{C_{\mathbf{f}}, C_{\hat{\mathbf{k}}}\right\}$ we have the thesis. □

Proposition 9.15. *With the same notation and assumptions of Lemma 9.14 it holds that*

$$\begin{aligned}
&\mathcal{L}_A\left(\mathbf{k}_{j+1}, \mathbf{f}_{j+1}, \hat{\mathbf{k}}_{j+1}, \hat{\mathbf{f}}_{j+1}; \boldsymbol{\xi}_{j+1}, \boldsymbol{\mu}_{j+1}\right) - \mathcal{L}_A\left(\mathbf{k}_j, \mathbf{f}_j, \hat{\mathbf{k}}_j, \hat{\mathbf{f}}_j; \boldsymbol{\xi}_j, \boldsymbol{\mu}_j\right) \\
&\leq \left(\frac{C^2}{\omega} - \frac{\rho}{2}\right)\left(\|\mathbf{f}_{j+1} - \mathbf{f}_j\|^2 + \left\|\hat{\mathbf{k}}_{j+1} - \hat{\mathbf{k}}_j\right\|^2\right) - \frac{\rho}{2}\left(\left\|\hat{\mathbf{f}}_{j+1} - \hat{\mathbf{f}}_j\right\|^2 + \|\mathbf{k}_{j+1} - \mathbf{k}_j\|^2\right).
\end{aligned}$$

Proof. We split the difference above as

$$\begin{aligned}
&\mathcal{L}_A\left(\mathbf{k}_{j+1}, \mathbf{f}_{j+1}, \hat{\mathbf{k}}_{j+1}, \hat{\mathbf{f}}_{j+1}; \boldsymbol{\xi}_{j+1}, \boldsymbol{\mu}_{j+1}\right) - \mathcal{L}_A\left(\mathbf{k}_j, f_j, \hat{\mathbf{k}}_j, \hat{\mathbf{f}}_j; \boldsymbol{\xi}_j, \boldsymbol{\mu}_j\right) \\
&= \mathcal{L}_A\left(\mathbf{k}_{j+1}, \mathbf{f}_{j+1}, \hat{\mathbf{k}}_{j+1}, \hat{\mathbf{f}}_{j+1}; \boldsymbol{\xi}_{j+1}, \boldsymbol{\mu}_{j+1}\right) - \mathcal{L}_A\left(\mathbf{k}_{j+1}, \mathbf{f}_{j+1}, \hat{\mathbf{k}}_{j+1}, \hat{\mathbf{f}}_{j+1}; \boldsymbol{\xi}_j, \boldsymbol{\mu}_j\right) \\
&+ \mathcal{L}_A\left(\mathbf{k}_{j+1}, \mathbf{f}_{j+1}, \hat{\mathbf{k}}_{j+1}, \hat{\mathbf{f}}_{j+1}; \boldsymbol{\xi}_j, \boldsymbol{\mu}_j\right) - \mathcal{L}_A\left(\mathbf{k}_j, \mathbf{f}_j, \hat{\mathbf{k}}_j, \hat{\mathbf{f}}_j; \boldsymbol{\xi}_j, \boldsymbol{\mu}_j\right).
\end{aligned} \tag{9.28}$$

Consider the first part

$$\begin{aligned}
&\mathcal{L}_A\left(\mathbf{k}_{j+1}, \mathbf{f}_{j+1}, \hat{\mathbf{k}}_{j+1}, \hat{\mathbf{f}}_{j+1}; \boldsymbol{\xi}_{j+1}, \boldsymbol{\mu}_{j+1}\right) - \mathcal{L}_A\left(\mathbf{k}_{j+1}, \mathbf{f}_{j+1}, \hat{\mathbf{k}}_{j+1}, \hat{\mathbf{f}}_{j+1}; \boldsymbol{\xi}_j, \boldsymbol{\mu}_j\right) \\
&= \left\langle \boldsymbol{\xi}_j - \boldsymbol{\xi}_{j+1}, \hat{\mathbf{f}}_{j+1} - \mathbf{f}_{j+1}\right\rangle + \left\langle \boldsymbol{\mu}_j - \boldsymbol{\mu}_{j+1}, \hat{\mathbf{k}}_{j+1} - \mathbf{k}_{j+1}\right\rangle \\
&= \frac{1}{\omega}\left\|\boldsymbol{\xi}_j - \boldsymbol{\xi}_{j+1}\right\|^2 + \frac{1}{\omega}\left\|\boldsymbol{\mu}_j - \boldsymbol{\mu}_{j+1}\right\|^2,
\end{aligned} \tag{9.29}$$

where the last step is obtained by recalling the definition of $\boldsymbol{\xi}_{j+1}$ and $\boldsymbol{\mu}_{j+1}$. We move to the second part.

As above we indicate with $\partial\mathcal{L}_A$ the subdifferential of L. Let a general element of $\partial\mathcal{L}_A$ be denoted by

$$\boldsymbol{\theta}\begin{pmatrix} \mathbf{f} \\ \hat{\mathbf{k}} \end{pmatrix} \in \partial\begin{pmatrix} \mathbf{f} \\ \hat{\mathbf{k}} \end{pmatrix} \mathcal{L}_A\left(\mathbf{k}_{j+1}, \mathbf{f}_{j+1}, \hat{\mathbf{k}}_{j+1}, \hat{\mathbf{f}}_{j+1}; \boldsymbol{\xi}_j, \boldsymbol{\mu}_j\right)$$

and

$$\boldsymbol{\theta}\begin{pmatrix} \hat{\mathbf{f}} \\ \mathbf{k} \end{pmatrix} \in \partial\begin{pmatrix} \hat{\mathbf{f}} \\ \mathbf{k} \end{pmatrix} \mathcal{L}_A\left(\mathbf{k}_{j+1}, \mathbf{f}_{j+1}, \hat{\mathbf{k}}_{j+1}, \hat{\mathbf{f}}_{j+1}; \boldsymbol{\xi}_j, \boldsymbol{\mu}_j\right).$$

Then it holds

$$
\begin{aligned}
&\mathcal{L}_A\left(\mathbf{k}_{j+1},\mathbf{f}_{j+1},\hat{\mathbf{k}}_{j+1},\hat{\mathbf{f}}_{j+1};\boldsymbol{\xi}_j,\boldsymbol{\mu}_j\right)-\mathcal{L}_A\left(\mathbf{k}_j,\mathbf{f}_j,\hat{\mathbf{k}}_j,\hat{\mathbf{f}}_j;\boldsymbol{\xi}_j,\boldsymbol{\mu}_j\right)\\
&=\mathcal{L}_A\left(\mathbf{k}_{j+1},\mathbf{f}_{j+1},\hat{\mathbf{k}}_{j+1},\hat{\mathbf{f}}_{j+1};\boldsymbol{\xi}_j,\boldsymbol{\mu}_j\right)-\mathcal{L}_A\left(\mathbf{k}_{j+1},\mathbf{f}_j,\hat{\mathbf{k}}_j,\hat{\mathbf{f}}_{j+1};\boldsymbol{\xi}_j,\boldsymbol{\mu}_j\right)\\
&+\mathcal{L}_A\left(\mathbf{k}_{j+1},\mathbf{f}_j,\hat{\mathbf{k}}_j,\hat{\mathbf{f}}_{j+1};\boldsymbol{\xi}_j,\boldsymbol{\mu}_j\right)-\mathcal{L}_A\left(\mathbf{k}_j,\mathbf{f}_j,\hat{\mathbf{k}}_j,\hat{\mathbf{f}}_j;\boldsymbol{\xi}_j,\boldsymbol{\mu}_j\right)\\
&\overset{(a)}{\leq}\left\langle\boldsymbol{\theta}_{\begin{pmatrix}\mathbf{f}\\ \hat{\mathbf{k}}\end{pmatrix}},\begin{pmatrix}\mathbf{f}_{j+1}\\ \hat{\mathbf{k}}_{j+1}\end{pmatrix}-\begin{pmatrix}\mathbf{f}_j\\ \hat{\mathbf{k}}_j\end{pmatrix}\right\rangle-\frac{\rho}{2}\left\|\begin{pmatrix}\mathbf{f}_{j+1}\\ \hat{\mathbf{k}}_{j+1}\end{pmatrix}-\begin{pmatrix}\mathbf{f}_j\\ \hat{\mathbf{k}}_j\end{pmatrix}\right\|^2\\
&+\left\langle\boldsymbol{\theta}_{\begin{pmatrix}\hat{\mathbf{f}}\\ \mathbf{k}\end{pmatrix}},\begin{pmatrix}\hat{\mathbf{f}}_{j+1}\\ \mathbf{k}_{j+1}\end{pmatrix}-\begin{pmatrix}\hat{\mathbf{f}}_j\\ \mathbf{k}_j\end{pmatrix}\right\rangle-\frac{\rho}{2}\left\|\begin{pmatrix}\hat{\mathbf{f}}_{j+1}\\ \mathbf{k}_{j+1}\end{pmatrix}-\begin{pmatrix}\hat{\mathbf{f}}_j\\ \mathbf{k}_j\end{pmatrix}\right\|^2\\
&\overset{(b)}{\leq}-\frac{\rho}{2}\left\|\begin{pmatrix}\mathbf{f}_{j+1}\\ \hat{\mathbf{k}}_{j+1}\end{pmatrix}-\begin{pmatrix}\mathbf{f}_j\\ \hat{\mathbf{k}}_j\end{pmatrix}\right\|^2-\frac{\rho}{2}\left\|\begin{pmatrix}\hat{\mathbf{f}}_{j+1}\\ \mathbf{k}_{j+1}\end{pmatrix}-\begin{pmatrix}\hat{\mathbf{f}}_j\\ \mathbf{k}_j\end{pmatrix}\right\|^2\\
&=-\frac{\rho}{2}\left(\|\mathbf{f}_{j+1}-\mathbf{f}_j\|^2+\left\|\hat{\mathbf{f}}_{j+1}-\hat{\mathbf{f}}_j\right\|^2+\|\mathbf{k}_{j+1}-\mathbf{k}_j\|^2+\left\|\hat{\mathbf{k}}_{j+1}-\hat{\mathbf{k}}_j\right\|^2\right),
\end{aligned}
\tag{9.30}
$$

where (a) follows from Assumption 9.3(iii) and (b) comes from the optimality condition for $\mathbf{k},\hat{\mathbf{k}},\hat{\mathbf{f}}$, and $\mathbf{f}$, i.e., from the fact that we can specialize the subgradients $\boldsymbol{\theta}$ to be the one which satisfies the optimality conditions.

Combining (9.29) and (9.30) with (9.28) and using Lemma 9.14, we obtain

$$
\begin{aligned}
&\mathcal{L}_A\left(\mathbf{k}_{j+1},\mathbf{f}_{j+1},\hat{\mathbf{k}}_{j+1},\hat{\mathbf{f}}_{j+1};\boldsymbol{\xi}_{j+1},\boldsymbol{\mu}_{j+1}\right)-\mathcal{L}_A\left(\mathbf{k}_j,\mathbf{f}_j,\hat{\mathbf{k}}_j,\hat{\mathbf{f}}_j;\boldsymbol{\xi}_j,\boldsymbol{\mu}_j\right)\\
&\leq\frac{1}{\omega}\left\|\boldsymbol{\xi}_j-\boldsymbol{\xi}_{j+1}\right\|^2+\frac{1}{\omega}\left\|\boldsymbol{\mu}_j-\boldsymbol{\mu}_{j+1}\right\|^2\\
&\quad-\frac{\rho}{2}\left(\|\mathbf{f}_{j+1}-\mathbf{f}_j\|^2+\left\|\hat{\mathbf{f}}_{j+1}-\hat{\mathbf{f}}_j\right\|^2+\|\mathbf{k}_{j+1}-\mathbf{k}_j\|^2+\left\|\hat{\mathbf{k}}_{j+1}-\hat{\mathbf{k}}_j\right\|^2\right)\\
&\leq\left(\frac{C^2}{\omega}-\frac{\rho}{2}\right)\left(\|\mathbf{f}_{j+1}-\mathbf{f}_j\|^2+\left\|\hat{\mathbf{k}}_{j+1}-\hat{\mathbf{k}}_j\right\|^2\right)-\frac{\rho}{2}\left(\left\|\hat{\mathbf{f}}_{j+1}-\hat{\mathbf{f}}_j\right\|^2+\|\mathbf{k}_{j+1}-\mathbf{k}_j\|^2\right)
\end{aligned}
$$

□

We are now in a position to prove that Algorithm 9.3 converges to a limit.

Lemma 9.16. *Let $\mathcal{L}_A$ be the functional defined in* (9.23) *and $\mathbf{k}_j,\mathbf{f}_j,\hat{\mathbf{k}}_j,\hat{\mathbf{f}}_j,\boldsymbol{\xi}_j,\boldsymbol{\mu}_j$ the iterates generated by Algorithm 9.3. Let Assumptions 9.1-9.3 and Conjecture 9.13 hold. Moreover, assume that $\frac{C^2}{\omega}-\frac{\rho}{2}<0$ we have that*

$$
\lim_{j\to\infty}\mathcal{L}_A\left(\mathbf{k}_j,\mathbf{f}_j,\hat{\mathbf{k}}_j,\hat{\mathbf{f}}_j;\boldsymbol{\xi}_j,\boldsymbol{\mu}_j\right)\geq\nu,
$$

where ν, defined in (9.5)*, is the global minimum of $J^{\delta,\epsilon}_{\alpha,\beta}(\mathbf{k},\mathbf{f})$.*

Proof. We observe that, since we assumed that $\frac{C^2}{\omega}-\frac{\rho}{2}<0$, from Proposition 9.15 it holds that the sequence $\left\{\mathcal{L}_A\left(\mathbf{k}_j,\mathbf{f}_j,\hat{\mathbf{k}}_j,\hat{\mathbf{f}}_j;\boldsymbol{\xi}_j,\boldsymbol{\mu}_j\right)\right\}_j$ is monotonically decreasing.

We now prove that the sequence is bounded from below. $\mathcal{L}_A$ can be rewritten as

$$\begin{aligned}
&\mathcal{L}_A\left(\mathbf{k}_j, \mathbf{f}_j, \hat{\mathbf{k}}_j, \hat{\mathbf{f}}_j; \boldsymbol{\xi}_j, \boldsymbol{\mu}_j\right) \\
&= \|B\left(\mathbf{k}_j, \mathbf{f}_j\right) - \mathbf{g}_\delta\|^2 + \alpha\left(\|\mathbf{f}_j\|^2 + \left\|\hat{\mathbf{f}}_j\right\|_{TV}\right) + \gamma\left\|\mathbf{k}_j - \mathbf{k}_\epsilon\right\|^2 + \beta\left\|\hat{\mathbf{k}}_j\right\|_{TV} \\
&+ \frac{\omega}{2}\left\|\hat{\mathbf{f}}_j - \mathbf{f}_j\right\|^2 - \left\langle \boldsymbol{\xi}_j, \hat{\mathbf{f}}_j - \mathbf{f}\right\rangle + \frac{\omega}{2}\left\|\hat{\mathbf{k}}_j - \mathbf{k}_j\right\|^2 - \left\langle \boldsymbol{\mu}_j, \hat{\mathbf{k}}_j - \mathbf{k}_j\right\rangle \\
&\in \|B\left(\mathbf{k}_j, \mathbf{f}_j\right) - \mathbf{g}_\delta\|^2 + \alpha\left(\|\mathbf{f}_j\|^2 + \left\|\hat{\mathbf{f}}_j\right\|_{TV}\right) + \gamma\left\|\mathbf{k}_j - \mathbf{k}_\epsilon\right\|^2 + \beta\left\|\hat{\mathbf{k}}_j\right\|_{TV} \\
&+ \frac{\omega}{2}\left\|\hat{\mathbf{f}}_j - \mathbf{f}_j\right\|^2 - \left\langle \nabla_{\mathbf{f}}\phi\left(\mathbf{k}_j, \mathbf{f}_j\right), \hat{\mathbf{f}}_j - \mathbf{f}_j\right\rangle + \frac{\omega}{2}\left\|\hat{\mathbf{k}}_j - \mathbf{k}_j\right\|^2 - \left\langle \partial\psi_\beta\left(\mathbf{k}_j\right), \hat{\mathbf{k}}_j - \mathbf{k}_j\right\rangle
\end{aligned}$$

. Using the fact that $\nabla_{\mathbf{f}}\phi$ and all the elements in $\partial\psi_\beta$ are Lipschitz continuous and considering that $\frac{C^2}{\omega} - \frac{\rho}{2} < 0$ by assumption we get

$$\begin{aligned}
&\mathcal{L}_A\left(\mathbf{k}_j, \mathbf{f}_j, \hat{\mathbf{k}}_j, \hat{\mathbf{f}}_j; \boldsymbol{\xi}_j, \boldsymbol{\mu}_j\right) \\
&\geq \|B\left(\mathbf{k}_j, \mathbf{f}_j\right) - \mathbf{g}_\delta\|^2 + \alpha\left(\|\mathbf{f}_j\|^2 + \|\mathbf{f}_j\|_{TV}\right) + \gamma\left\|\mathbf{k}_j - \mathbf{k}_\epsilon\right\|^2 + \beta\left\|\mathbf{k}_j\right\|_{TV} \\
&\geq \nu
\end{aligned}$$

where in the last step we have used the fact that ν is the global minimum of $J^{\delta,\epsilon}_{\alpha,\beta}(\mathbf{k}, \mathbf{f})$.

Since the sequence $\left\{\mathcal{L}_A\left(\mathbf{k}_j, \mathbf{f}_j, \hat{\mathbf{k}}_j, \hat{\mathbf{f}}_j; \boldsymbol{\xi}_j, \boldsymbol{\mu}_j\right)\right\}_j$ is monotonically decreasing and bounded, we have that it converges. □

We can now prove our primary result.

Theorem 9.17. *Let* $\mathbf{p}_* = \left(\mathbf{k}_*, \mathbf{f}_*, \hat{\mathbf{k}}_*, \hat{\mathbf{f}}_*, \boldsymbol{\xi}_*, \boldsymbol{\mu}_*\right)$ *be the limit point of Algorithm 9.3. Assume that Assumptions 9.1–9.3 hold and that Conjecture 9.13 is satisfied. Then the following hold*

(a) $\mathbf{p}_*$ *is a stationary point, that is*

(i) $\mathbf{f}_* = \hat{\mathbf{f}}_*$ *and* $\mathbf{k}_* = \hat{\mathbf{k}}_*$*;*

(ii) $\mathbf{0} = \nabla_{\mathbf{f}}\phi\left(\mathbf{f}_*, \mathbf{k}_*\right) + \boldsymbol{\xi}_*$ *and* $\mathbf{0} = \nabla_{\mathbf{k}}\phi\left(\mathbf{f}_*, \mathbf{k}_*\right) + \boldsymbol{\mu}_*$*;*

(iii) $\left(\hat{\mathbf{k}}_*, \hat{\mathbf{f}}_*\right) \in \arg\min_{(\hat{\mathbf{k}}, \hat{\mathbf{f}})} \alpha\left\|\hat{\mathbf{f}}\right\|_{TV} + \left\langle \mathbf{f}_* - \hat{\mathbf{f}}, \boldsymbol{\xi}_*\right\rangle + \beta\left\|\hat{\mathbf{k}}\right\|_{TV} + \left\langle \mathbf{k}_* - \hat{\mathbf{k}}, \boldsymbol{\mu}_*\right\rangle$.

(b) Assume now that $\Omega_{\mathbf{f}} \times \Omega_{\mathbf{k}}$ *is compact then*

$$\lim_{j\to\infty} \operatorname{dist}\left(\left(\mathbf{f}_j, \mathbf{k}_j, \hat{\mathbf{f}}_j, \hat{\mathbf{k}}_j; \boldsymbol{\xi}_j, \boldsymbol{\mu}_j\right), Z^*\right) = 0,$$

where Z^* *denotes the set of stationary points and* dist *the Euclidean distance between sets and points.*

Proof. We only prove part (a), we omit the proof of part (b) since it can be copied with no significant modification from [87, Theorem 2.4 part 3].

From Proposition 9.15 and Lemma 9.16 for $j \to \infty$ we have that

$$\|\mathbf{f}_{j+1} - \mathbf{f}_j\| \to 0, \quad \left\|\hat{\mathbf{f}}_{j+1} - \hat{\mathbf{f}}_j\right\| \to 0, \quad \|\mathbf{k}_{j+1} - \mathbf{k}_j\| \to 0, \quad \left\|\hat{\mathbf{k}}_{j+1} - \hat{\mathbf{k}}_j\right\| \to 0.$$

Moreover, in force of Lemma 9.14 it holds

$$\left\|\boldsymbol{\xi}_{j+1}-\boldsymbol{\xi}_j\right\| \to 0, \quad \left\|\boldsymbol{\mu}_{j+1}-\boldsymbol{\mu}_j\right\| \to 0. \tag{9.31}$$

Let ϕ be the functional defined in (9.24) and recall that $\mathbf{p}^* = \left(\mathbf{f}_*, \mathbf{k}_*, \hat{\mathbf{f}}_*, \hat{\mathbf{k}}_*, \boldsymbol{\xi}_*, \boldsymbol{\mu}_*\right)$ is the limit point (that exists in virtue of Lemma 9.16) generated by the iterations of Algorithm 9.3.

Observe that, from (9.31), it follows that

$$\mathbf{f}_* = \hat{\mathbf{f}}_*, \quad \mathbf{k}_* = \hat{\mathbf{k}}_*,$$

which proves (i).

We move now to the proof of (ii).

From (9.27) we have that $\mathbf{0} = \nabla_{\mathbf{f}}\phi\left(\mathbf{f}_{j+1}, \mathbf{k}_{j+1}\right) + \boldsymbol{\xi}_{j+1}$ and by taking the limit for $j \to \infty$ we get that

$$\mathbf{0} = \nabla_{\mathbf{f}}\phi\left(\mathbf{f}_*, \mathbf{k}_*\right) + \boldsymbol{\xi}_*.$$

Consider now $\mathbf{k}_{j+1}$, imposing the optimality condition for $\mathbf{k}_{j+1}$ to (9.23), it holds

$$\begin{aligned}
\mathbf{0} &= \nabla_{\mathbf{k}}\phi\left(\mathbf{f}_j, \mathbf{k}_{j+1}\right) + \boldsymbol{\mu}_j - \omega\left(\hat{\mathbf{k}}_j - \mathbf{k}_{j+1}\right) \\
&= \nabla_{\mathbf{k}}\phi\left(\mathbf{f}_j, \mathbf{k}_{j+1}\right) + \boldsymbol{\mu}_j - \omega\left(\hat{\mathbf{k}}_j - \mathbf{k}_{j+1}\right) - \omega\hat{\mathbf{k}}_{j+1} + \omega\hat{\mathbf{k}}_{j+1} \\
&= \nabla_{\mathbf{k}}\phi\left(\mathbf{f}_j, \mathbf{k}_{j+1}\right) + \boldsymbol{\mu}_{j+1} - \omega\left(\hat{\mathbf{k}}_j - \hat{\mathbf{k}}_{j+1}\right),
\end{aligned}$$

where we added and subtracted $\omega\hat{\mathbf{k}}_{j+1}$ in order to make $\boldsymbol{\mu}_{j+1}$ appear.

Taking the limit for $j \to \infty$, since $\left\|\hat{\mathbf{k}}_{j+1} - \hat{\mathbf{k}}_j\right\| \to 0$, we have that

$$\mathbf{0} = \nabla_{\mathbf{k}}\phi\left(\mathbf{f}_*, \mathbf{k}_*\right) + \boldsymbol{\mu}_*,$$

completing the proof of (ii).

For the proof of (iii) consider the optimality condition of (9.23) for $\hat{\mathbf{f}}_{j+1}$, we have that there exists $\boldsymbol{\theta} \in \partial\psi_\alpha\left(\hat{\mathbf{f}}_{j+1}\right)$, where ψ_α is defined in (9.25), such that

$$\left\langle \hat{\mathbf{f}} - \hat{\mathbf{f}}_{j+1}, \boldsymbol{\theta} - \left(\boldsymbol{\xi}_j - \omega\left(\hat{\mathbf{f}}_{j+1} - \hat{\mathbf{f}}_j\right)\right)\right\rangle \geq 0 \quad \forall\, \hat{\mathbf{f}}.$$

By convexity of ψ_α we have that

$$\psi_\alpha\left(\hat{\mathbf{f}}\right) - \psi_\alpha\left(\hat{\mathbf{f}}_{j+1}\right) + \left\langle \hat{\mathbf{f}} - \hat{\mathbf{f}}_{j+1}, -\boldsymbol{\xi}_j - \omega\left(\hat{\mathbf{f}}_{j+1} - \hat{\mathbf{f}}_j\right)\right\rangle \geq 0 \quad \forall\, \hat{\mathbf{f}}.$$

Taking to the limit for $j \to \infty$ we have that

$$\psi_\alpha\left(\hat{\mathbf{f}}\right) - \psi_f\left(\hat{\mathbf{f}}_*\right) - \left\langle \hat{\mathbf{f}} - \hat{\mathbf{f}}_*, \boldsymbol{\xi}_*\right\rangle \geq 0 \quad \forall \hat{\mathbf{f}}.$$

Using the fact that $\hat{\mathbf{f}}_* = \mathbf{f}_*$ we have

$$\psi_\alpha\left(\hat{\mathbf{f}}\right) + \left\langle \mathbf{f}_* - \hat{\mathbf{f}}, \boldsymbol{\xi}_*\right\rangle - \left(\psi_\alpha\left(\hat{\mathbf{f}}_*\right) + \left\langle \mathbf{f}_* - \hat{\mathbf{f}}_*, \boldsymbol{\xi}_*\right\rangle\right) \geq 0 \quad \forall\, \hat{\mathbf{f}}.$$

Similarly we can prove, recalling the definition of ψ_β in (9.26), that

$$\psi_\beta\left(\hat{\mathbf{k}}\right) + \left\langle \mathbf{k}_* - \hat{\mathbf{k}}, \boldsymbol{\mu}_* \right\rangle - \left(\psi_\beta\left(\hat{\mathbf{k}}_*\right) + \left\langle \mathbf{k}_* - \hat{\mathbf{k}}_*, \boldsymbol{\mu}_* \right\rangle\right) \geq 0 \quad \forall\, \hat{\mathbf{k}},$$

which concludes the proof of (iii). We have then proven that the limit point of the sequence is a stationary point for the unconstrained problem. □

9.4 Numerical examples

We now give some numerical examples. Firstly we reformulate our algorithm to be more efficient, then we compare our approach with the one in [18] on the same example they proposed in their work. Finally we show the effectiveness of the proposed method on other situations.

We consider the framework of image deblurring with spatially invariant blur, in this setting $\mathbf{k}$ will be the PSF. As we stated in Chapter 2 if we fix one of the two variables and discretize (2.5) by a collocation method, we get a linear system which is severely ill-conditioned. We call $A_{\mathbf{k}}$ the discretization of $B(k, \cdot)$ and $A_{\mathbf{f}}$ the discretized version of $B(\cdot, f)$. For simplicity we assume that both the image and the PSF are periodic and so $A_{\mathbf{f}}$ and $A_{\mathbf{k}}$ are BCCB matrices. Since we imposed periodic boundary condition on D defined in (9.20) we have that D is a BCCB matrix as well. Thanks to this choice all the matrices involved are diagonalizable using the Fourier transform and so all the linear systems involved can be solved exactly using the FFT in $O(n \log n)$ flops.

Both the image and the PSF should not have negative values, thus we want to constrain them inside the nonnegative cone. Since the assumptions of Section 9.2 hold, we can use the flux conservation on $\mathbf{f}$. Moreover, since we know that $\mathbf{k}$ should sum up to 1, we want to constrain the flux of $\mathbf{k}$ to be 1. Summarizing we set

$$\Omega_{\mathbf{f}} = \Omega_0 \cap \Omega_F,$$

$$\Omega_{\mathbf{k}} = \Omega_0 \cap \Omega_1,$$

where Ω_0 is the nonnegative cone, Ω_F is defined in (9.10), and $\Omega_1 = \{\mathbf{k} \in \mathbb{R}^{n\times n} \mid \mathsf{flux}(\mathbf{k}) = 1\}$.

Projecting on either $\Omega_{\mathbf{f}}$ or $\Omega_{\mathbf{k}}$ is not trivial, then we have to resort to the technique described in Algorithm 9.1 for decoupling the projection on the components of $\Omega_{\mathbf{f}}$ and $\Omega_{\mathbf{k}}$. By doing so we are able to perform the projections into the nonnegative cone, on Ω_1, and on Ω_F in $O(n)$ flops. We are then actually able to introduce both the nonnegativity and the flux conservation constraints in our method.

For the evaluation of the performances of the method we use the Signal to Noise Ratio (SNR), which is computed as

$$\mathrm{SNR}(\mathbf{x}) = 20 \log_{10}\left(\frac{\|\mathbf{x}^\dagger\|}{\|\mathbf{x} - \mathbf{x}^\dagger\|}\right).$$

The discussion on how to choose the appropriate α and β is out of the scope of this chapter and thus we choose the optimal one, i.e., the one which gives the highest value of SNR among some tested ones.

For stopping both methods we use the relative distance between two consecutive iterations, i.e., we stop as soon as

$$\frac{\|\mathbf{f}_{j-1} - \mathbf{f}_j\|}{\|\mathbf{f}_{j-1}\|} \leq 10^{-4}.$$

Since in the considered examples we use the flux conservation (9.9) and nonnegativity constraints, we have that the bounds in Conjecture 9.13 are

$$\varphi_k = 1 \qquad \text{and} \qquad \varphi_f = \mathsf{flux}\left(\mathbf{g}_\delta\right). \tag{9.32}$$

In the computed examples we are going to show that, at least experimentally, the norm of the iterates are bounded by the quantities in (9.32) and so that Conjecture 9.13 holds.

We want to show that the introduction of the knowledge of the presence of the noise in the PSF and the flux constraint helps in getting better reconstructions. We confront SeB-A against the deconvolution obtained using directly the noisy PSF, i.e., the reconstruction obtained by minimizing

$$\mathbf{f} = \arg\min_{\mathbf{f}\in\Omega_\mathbf{f}} \|B\left(\mathbf{k}_\epsilon, \mathbf{f}\right) - \mathbf{g}_\delta\|^2 + \alpha\left(\|\mathbf{f}\|_{TV} + \|\mathbf{f}\|^2\right), \tag{9.33}$$

where, for the sake of simplicity, we consider only one auxiliary variable for the constraint on $\Omega_\mathbf{f}$. This algorithm is very similar to the one proposed in [38], the only difference is the presence of $\|\mathbf{f}\|^2$ in the regularization term. To minimize (9.33) we proceed as in [38], i.e., decoupling the variables and forming the Augmented Lagrangian and then using ADMM.

The decoupled problem in finite dimension becomes

$$\mathbf{f} = \arg\min_{\tilde{\mathbf{f}}\in\Omega_\mathbf{f},\hat{\mathbf{f}},\mathbf{f}} \left\{ \|B\left(\mathbf{k}_\epsilon, \mathbf{f}\right) - \mathbf{g}_\delta\|^2 + \alpha\left(\sum_{i=1}^{N}\left\|\hat{\mathbf{f}}_i\right\| + \|\mathbf{f}\|^2\right), \right.$$
$$\left. \tilde{\mathbf{f}} = \mathbf{f},\ \hat{\mathbf{f}}_i = D_i\mathbf{f},\ \forall i = 1,\ldots,N \right\}.$$

The related Augmented Lagrangian is

$$\mathcal{L}_A\left(\tilde{\mathbf{f}}, \hat{\mathbf{f}}, \mathbf{f}, \boldsymbol{\lambda}, \boldsymbol{\xi}\right) = \|B\left(\mathbf{k}_\epsilon, \mathbf{f}\right) - \mathbf{g}_\delta\|^2 + \alpha\left(\sum_{i=1}^{N}\left\|\hat{\mathbf{f}}_i\right\| + \|\mathbf{f}\|^2\right)$$
$$+ \frac{\omega_1}{2}\left\|\tilde{\mathbf{f}} - \mathbf{f}\right\|^2 - \lambda^t\left(\tilde{\mathbf{f}} - \mathbf{f}\right) + \sum_{i=1}^{N}\left[\frac{\omega_2}{2}\left\|\left(\hat{\mathbf{f}}\right)_i - D_i\mathbf{f}\right\|^2 - \left(\boldsymbol{\xi}\right)_i^t\left(\left(\hat{\mathbf{f}}\right)_i - D_i\mathbf{f}\right)\right].$$

The ADMM algorithm becomes

Algorithm 9.4 (Tikhonov-TV)**.** *Let* $\mathbf{f}_0$, $\boldsymbol{\lambda}_0$, *and* $\boldsymbol{\xi}_0$ *be initial guesses for* $\mathbf{f}$, $\boldsymbol{\lambda}$, *and* $\boldsymbol{\xi}$, *respectively. Let* $\omega_1, \omega_2 > 0$ *be real fixed positive numbers.*

for $j = 0, 1, \ldots$

$$\begin{pmatrix} \tilde{\mathbf{f}}_{j+1} \\ \hat{\mathbf{f}}_{j+1} \end{pmatrix} = \arg\min_{\tilde{\mathbf{f}},\hat{\mathbf{f}}} \mathcal{L}_A\left(\tilde{\mathbf{f}}, \hat{\mathbf{f}} \middle| \mathbf{f}_j; \boldsymbol{\lambda}_j, \boldsymbol{\xi}_j\right)$$
$$\mathbf{f}_{j+1} = \arg\min_{\mathbf{f}} \mathcal{L}_A\left(\mathbf{f} \middle| \tilde{\mathbf{f}}_{j+1}, \hat{\mathbf{f}}_{j+1}; \boldsymbol{\lambda}_j, \boldsymbol{\xi}_j\right)$$
$$\boldsymbol{\lambda}_{j+1} = \boldsymbol{\lambda}_j - \omega_1\left(\tilde{\mathbf{f}}_{j+1} - \mathbf{f}_{j+1}\right)$$
$$\boldsymbol{\xi}_{j+1} = \boldsymbol{\xi}_j - \omega_2\left(\hat{\mathbf{f}}_{j+1} - D\mathbf{f}_{j+1}\right)$$

end

The various minimizations in the algorithm above have a closed form, see [38] and Algorithm 9.5.

Since the minimized functional in (9.33) is convex it admits an unique minimizer and thus Algorithm 9.4 converges to it in force of the classical ADMM theory.

The implementation of SeB-A Algorithm 9.2 can be quite expensive. In fact, since there is no closed form for the minimization with respect to the auxiliary variables $\hat{\mathbf{k}}$ and $\hat{\mathbf{f}}$, the usage of some iterative method is required. This implies that the computational cost of SeB-A can be fairly high. In order to damp this cost we now present a different implementation of the algorithm. We will refer to this method as Computational SeB-A (CSeB-A). This algorithm does not require any inner cycle for the solution of the intermediate problems, however, we are not able to present a rigorous result of convergence. In the first example we are going to show that SeB-A and CSeB-A give equivalent results and thus in the following experiments we are going to use only the latter for computational convenience.

Let us rewrite (9.19) explicitly and reformulate as in [38]

$$\begin{aligned}
(\mathbf{k}^*, \mathbf{f}^*) &= \arg \min_{\mathbf{k}\in\Omega_{\mathbf{k}}, \mathbf{f}\in\Omega_{\mathbf{f}}} \|B(\mathbf{k},\mathbf{f}) - \mathbf{g}_\delta\|^2 + \alpha \left(\|\mathbf{f}\|^2 + \sum_{i=1}^N \|D_i \mathbf{f}\| \right) \\
&\quad + \gamma \|\mathbf{k} - \mathbf{k}_\epsilon\|^2 + \beta \sum_{i=1}^N \|D_i \mathbf{k}\| \\
&= \arg \min_{\substack{\tilde{\mathbf{k}}\in\Omega_{\mathbf{k}}, \tilde{\mathbf{f}}\in\Omega_{\mathbf{f}} \\ \hat{\mathbf{k}}, \hat{\mathbf{f}}, \mathbf{k}, \mathbf{f}}} \left\{ \|B(\mathbf{k},\mathbf{f}) - \mathbf{g}_\delta\|^2 + \alpha \left(\|\mathbf{f}\|^2 + \sum_{i=1}^N \left\|\hat{\mathbf{f}}_i\right\| \right) \right. \\
&\quad + \gamma \|\mathbf{k} - \mathbf{k}_\epsilon\|^2 + \beta \sum_{i=1}^N \left\|\hat{\mathbf{k}}_i\right\|, \\
&\quad \left. \mathbf{k} = \tilde{\mathbf{k}}, \mathbf{f} = \tilde{\mathbf{f}}, D_i \mathbf{k} = \hat{\mathbf{k}}_i, D_i \mathbf{f} = \hat{\mathbf{f}}_i,\ i = 1, \dots, N \right\},
\end{aligned}$$

where, $\mathbf{f}, \tilde{\mathbf{f}}, \mathbf{k}, \tilde{\mathbf{k}} \in \mathbb{R}^N$, $\hat{\mathbf{f}}, \hat{\mathbf{k}} \in \mathbb{R}^{N\times 2}$ and we have $\left(\hat{\mathbf{f}}\right)_i = \begin{pmatrix} \hat{\mathbf{f}}_{i,1} \\ \hat{\mathbf{f}}_{i,2} \end{pmatrix}$ and similarly for $\left(\hat{\mathbf{k}}\right)_i$.

We now write the Augmented Lagrangian of the minimization above

$$\begin{aligned}
&\mathcal{L}_A\left(\tilde{\mathbf{f}}, \hat{\mathbf{f}}, \mathbf{f}, \tilde{\mathbf{k}}, \hat{\mathbf{k}}, \mathbf{k}; \boldsymbol{\lambda}, \boldsymbol{\xi}, \boldsymbol{\zeta}, \boldsymbol{\mu}\right) \\
&= \|B(\mathbf{k},\mathbf{f}) - \mathbf{g}_\delta\|^2 + \alpha \left(\|\mathbf{f}\|^2 + \sum_{i=1}^N \left\| \left(\hat{\mathbf{f}}\right)_i \right\| \right) + \gamma \|\mathbf{k} - \mathbf{k}_\epsilon\| + \beta \sum_{i=1}^N \left\| \left(\hat{\mathbf{k}}\right)_i \right\| \\
&+ \frac{\omega_1}{2} \left\| \tilde{\mathbf{f}} - \mathbf{f} \right\|^2 - \left\langle \lambda, \tilde{f} - \mathbf{f} \right\rangle + \sum_{i=1}^N \left[\frac{\omega_2}{2} \left\| \left(\hat{\mathbf{f}}\right)_i - D_i \mathbf{f} \right\|^2 - \left\langle (\boldsymbol{\xi})_i, \left(\hat{\mathbf{f}}\right)_i - D_i \mathbf{f} \right\rangle \right] \\
&+ \frac{\omega_3}{2} \left\| \tilde{\mathbf{k}} - \mathbf{k} \right\|^2 - \langle \boldsymbol{\zeta}, ktb - k_\epsilon \rangle + \sum_{i=1}^N \left[\frac{\omega_4}{2} \left\| \left(\hat{\mathbf{k}}\right)_i - D_i \mathbf{k} \right\|^2 - \left\langle (\boldsymbol{\mu})_i, \left(\hat{\mathbf{k}}\right)_i - D_i \mathbf{k} \right\rangle \right],
\end{aligned}$$

where $\boldsymbol{\lambda}, \boldsymbol{\zeta} \in \mathbb{R}^N$ and $\boldsymbol{\xi}, \boldsymbol{\mu} \in \mathbb{R}^{N\times 2}$.

We can apply the ADMM algorithm obtaining

Algorithm 9.5 (CSeB-A). *Given* $\mathbf{f}_0$, $\mathbf{k}_0$, $\boldsymbol{\lambda}_0$, $\boldsymbol{\xi}_0$, $\boldsymbol{\zeta}_0$ *and* $\boldsymbol{\mu}_0$ *initial guesses for* $\mathbf{f}$, $\mathbf{k}$, $\boldsymbol{\lambda}$, $\boldsymbol{\xi}$, $\boldsymbol{\zeta}$, *and* $\boldsymbol{\mu}$, *respectively. Let* $\omega_1, \omega_2, \omega_3, \omega_4 > 0$ *be real fixed numbers.*

for $j = 0, 1, \ldots$

$$\begin{pmatrix} \tilde{\mathbf{f}}_{j+1} \\ \hat{\mathbf{f}}_{j+1} \\ \mathbf{k}_{j+1} \end{pmatrix} = \arg\min_{\tilde{\mathbf{f}},\hat{\mathbf{f}},\mathbf{k}} \mathcal{L}_A\left(\tilde{\mathbf{f}}, \hat{\mathbf{f}}, \mathbf{k} \middle| \tilde{\mathbf{k}}_j, \hat{\mathbf{k}}_j, \mathbf{f}_j; \boldsymbol{\lambda}_j, \boldsymbol{\xi}_j, \boldsymbol{\zeta}_j, \boldsymbol{\mu}_j\right)$$

$$\begin{pmatrix} \tilde{\mathbf{k}}_{j+1} \\ \hat{\mathbf{k}}_{j+1} \\ \mathbf{f}_{j+1} \end{pmatrix} = \arg\min_{\tilde{\mathbf{k}},\hat{\mathbf{k}},\mathbf{f}} \mathcal{L}_A\left(\tilde{\mathbf{k}}, \hat{\mathbf{k}}, \mathbf{f} \middle| \tilde{\mathbf{f}}_{j+1}, \hat{\mathbf{f}}_{j+1}, \mathbf{k}_{j+1}; \boldsymbol{\lambda}_j, \boldsymbol{\xi}_j, \boldsymbol{\zeta}_j, \boldsymbol{\mu}_j\right)$$

$$\boldsymbol{\lambda}_{j+1} = \boldsymbol{\lambda}_j - \omega_1\left(\tilde{\mathbf{f}}_{j+1} - \mathbf{f}_{j+1}\right)$$

$$\boldsymbol{\xi}_{j+1} = \xi_j - \omega_2\left(\hat{\mathbf{f}}_{j+1} - D\mathbf{f}_{j+1}\right)$$

$$\boldsymbol{\zeta}_{j+1} = \boldsymbol{\zeta}_j - \omega_3\left(\tilde{\mathbf{k}}_{j+1} - \mathbf{k}_{j+1}\right)$$

$$\boldsymbol{\mu}_{j+1} = \boldsymbol{\mu}_j - \omega_4\left(\hat{\mathbf{k}}_{j+1} - D\mathbf{k}_{j+1}\right)$$

end

As we stated above, thanks to Assumption 9.2 and to the fact that all the matrices involved are BCCB matrices, the minimization above are easily computed and all have a closed form.

$$\tilde{\mathbf{f}}_{j+1} = P_{\Omega_{\mathbf{f}}}\left(\mathbf{f}_j + \frac{\boldsymbol{\lambda}_j}{\omega_1}\right)$$

$$\left(\hat{\mathbf{f}}_{j+1}\right)_i = \frac{\left(D_i\mathbf{f}_j + \frac{1}{\omega_2}\left(\boldsymbol{\xi}_j\right)_i\right)}{\left\|D_i\mathbf{f}_j + \frac{1}{\omega_2}\left(\boldsymbol{\xi}_j\right)_i\right\|} \circ \left(\left\|D_i\mathbf{f}_j + \frac{1}{\omega_2}\left(\boldsymbol{\xi}_j\right)_i\right\| - \frac{\alpha}{\omega_2}\right)_+$$

$$\mathbf{k}_{j+1} = \left(2A_{\mathbf{f}_j}^* A_{\mathbf{f}_j} + 2\gamma I + \omega_3 I + \omega_4 D^* D\right)^{-1}\left(2A_{\mathbf{f}_j}^*\mathbf{g}_\delta + 2\gamma\mathbf{k}_\epsilon - \boldsymbol{\zeta}_j + \omega_3\tilde{\mathbf{k}}_j - D^*\boldsymbol{\mu}_j + \omega_4 D^*\hat{\mathbf{k}}_j\right)$$

$$\tilde{\mathbf{k}}_{j+1} = P_{\Omega_{\mathbf{k}}}\left(\mathbf{k}_{j+1} + \frac{\boldsymbol{\zeta}_j}{\omega_3}\right)$$

$$\left(\hat{\mathbf{k}}_{j+1}\right)_i = \frac{\left(D_i\mathbf{k}_{j+1} + \frac{1}{\omega_4}\left(\boldsymbol{\mu}_j\right)_i\right)}{\left\|D_i\mathbf{k}_{j+1} + \frac{1}{\omega_4}\left(\boldsymbol{\mu}_j\right)_i\right\|} \circ \left(\left\|D_i\mathbf{k}_{j+1} + \frac{1}{\omega_2}\left(\boldsymbol{\mu}_j\right)_i\right\| - \frac{\beta}{\omega_4}\right)_+$$

$$\mathbf{f}_{j+1} = \left(2A_{\mathbf{k}_{j+1}}^* A_{\mathbf{k}_{j+1}} + 2\alpha I + \omega_1 I + \omega_2 D^* D\right)^{-1}\left(2A_{\mathbf{k}_{j+1}}^*\mathbf{g}_\delta - \boldsymbol{\lambda} + \omega_1\tilde{\mathbf{f}}_{j+1} - D^*\boldsymbol{\xi}_j + \omega_2 D^*\hat{\mathbf{f}}_{j+1}\right)$$

We can now proceed with the numerical tests. The first example was performed on a laptop pc with an Intel Core i7 6700HQ with 16GB of RAM running MATLAB 2016a 64-bit. The other tests (a part the one from [18]) were made on another laptop pc with an Intel Core i5 3337U with 6GB of RAM running MATLAB 2015a 64-bit on Windows 10.

Equivalence between SeB-A and CSeB-A We would like now to show that SeB-A and CSeB-A give equivalent results. For this purpose we use a relatively small example, since the images involved are of 128×128 pixels. We consider the image deblurring problem

(a) (b) (c) (d)

FIGURE 9.1: Boat test problem: (a) Test image, (b) Noise-free PSF, (c) Blurred and noisy image, (d) Noisy PSF.

in Figure 9.1. We blur the image in Figure 9.1(a) with an out of focus PSF then add white Gaussian noise, such that $\xi = 0.02$. We add 70% of Gaussian noise to the PSF $\mathbf{k}$ to obtain $\mathbf{k}_\epsilon$.

We run both SeB-A and CSeB-A on several choices of α and β. In particular we test the method on a 7×7 grid of parameters logarithmically spaced between 10^{-4} and 10^{-2}. In Figure 9.2 we show the RREs obtained with different choices of α and β with the two methods. We can see that the errors are almost the same. Moreover, we can see that the errors obtained with CSeB-A are slightly better than the one obtained with SeB-A. This can be due to the fact that in SeB-A we approximate the minimization with respect to $\hat{\mathbf{f}}$ and $\hat{\mathbf{k}}$ whereas in CSeB-A all the minimization are exact.

We can also observe that changing α does not affect too much the quality of the reconstruction of $\mathbf{k}$ and, similarly, changing β does not affect to much the quality of the reconstruction of $\mathbf{f}$.

We conclude by showing, in Figure 9.3, the best restorations of $\mathbf{k}$ and $\mathbf{f}$ for both methods. From both visual inspection and the comparison of the resulting SNRs we can see that the difference between the two methods in term of accuracy is very small.

Comparison with dbl-RTLS We now compare our approach to the one in [18] on the same example proposed in that work. In Figure 9.4 we show the true image, the PSF, and the noise-free blurred image, we add different level of noise to both the image and the PSF and analyze the behavior in each situation. In Table 9.1 we show the comparison of the SNR with the different levels of noise and we also show the used parameter α and β for both CSeB-A and Tikhonov-TV. Finally, in Figure 9.5 we can see the reconstructions.

From these comparisons we can see that the proposed approach is able to get a better restoration of the image and in particular of the PSF. The gap between the two approaches gets more and more evident as the quantity of noise increases.

Finally we want to show that the norm of the iterates remains bounded and that it holds

$$\|\mathbf{k}_j\| \leq 1 \text{ and } \|\mathbf{f}_j\| \leq \mathsf{flux}(\mathbf{g}_\delta) \ \forall j = 1, 2, \dots .$$

In Figure 9.10 we show the norm of the iterates compared with the above bounds in the case with 8% of noise. We can see that the norm of the iterates stabilizes in very few iterations and that it is several order of magnitude smaller than the proposed bound.

Satellite We now test our method on the Satellite image blurred with an atmospheric PSF. We add white Gaussian noise such that $\delta = 0.05\,\|g\|$, i.e., with $\xi = 0.05$, and we also add

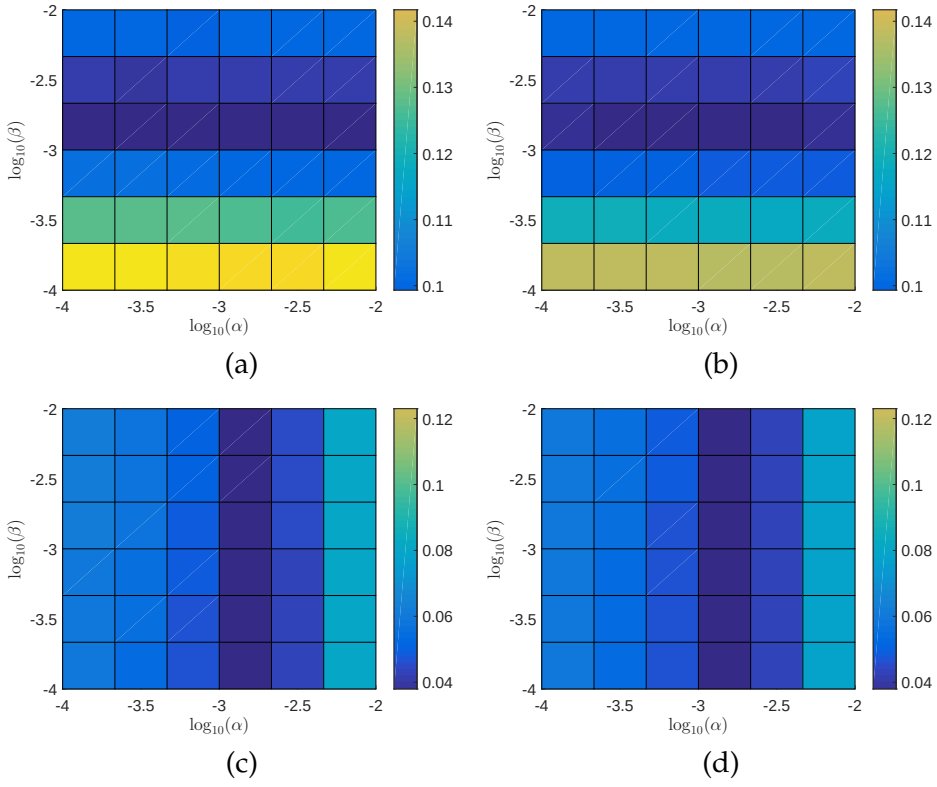
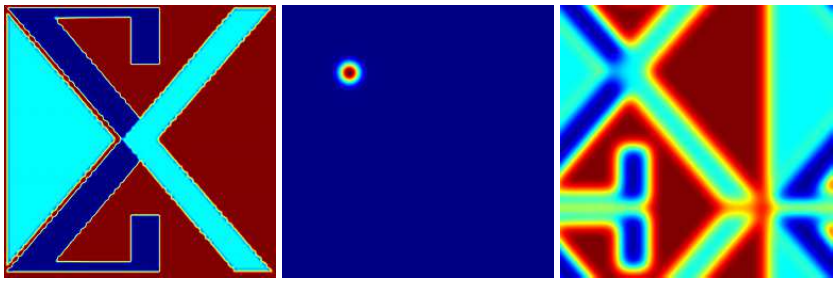


FIGURE 9.2: Boat test problem errors comparison: Errors comparison between SeB-A and CSeB-A against α and β. (a) RRE of the image for SeB-A, (b) RRE of the image for CSeB-A, (c) RRE of the PSF for SeB-A, (d) RRE of the image for CSeB-A.

FIGURE 9.3: Boat test problem optimal reconstructions: (a) PSF computed with SeB-A (SNR=29.129), (b) PSF computed with CSeB-A (SNR=29.211), (c) Image computed with SeB-A (SNR=20.886), (d) Image computed with CSeB-A (SNR=20.859).

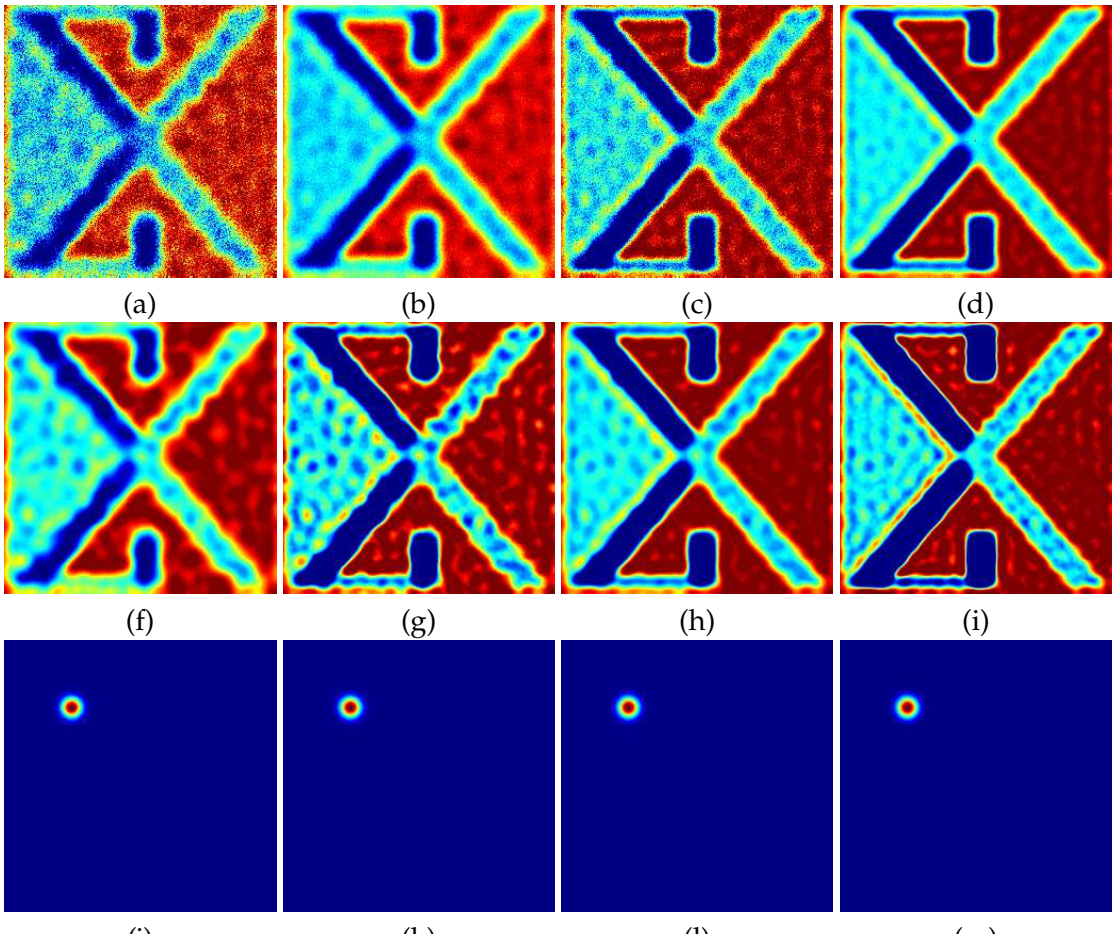
Noise Level	SNR f			SNR k	
	dbl-RTLS	Tikhonov-TV	CSeB-A	dbl-RTLS	CSeB-A
8%	8.627	9.516	12.481	11.679	23.257
4%	12.116	12.116	13.882	13.638	23.870
2%	13.099	12.891	15.039	15.041	23.958
1%	15.190	15.086	15.830	15.997	24.141

TABLE 9.1: Example from [18]: Confront of the SNR obtained with the method described in [18] (dbl-RTLS), Algorithm 9.4 (Tikhonov-TV) and our (computational) proposal (CSeB-A). Note that Tikhonov-TV does not take into account the noise in the PSF.



(a) (b) (c)

FIGURE 9.4: Example from [18]: (a) Test image, (b) PSF, (c) Blurred image (without noise).

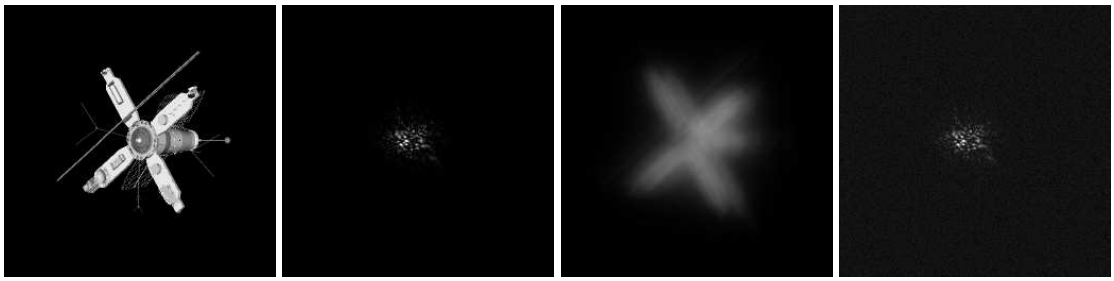


(a) (b) (c) (d)

(f) (g) (h) (i)

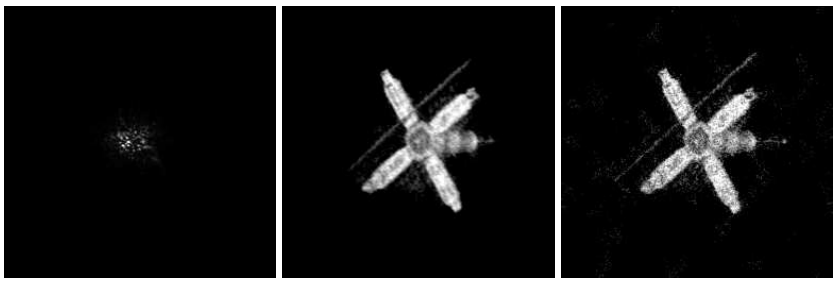
(j) (k) (l) (m)

FIGURE 9.5: Example from [18] reconstructions with different noise levels: on the first row the reconstructed images with Tikhonov-TV, on the second and third row the reconstructed images and PSF with CSeB-A, respectively. From left to right with 8%, 4%, 2%, and 1% of noise.



(a) (b) (c) (d)

FIGURE 9.6: Satellite test problem: (a) Test image, (b) Noise-free PSF, (c) Blurred and noisy image, (d) Noisy PSF.



(a) (b) (c)

FIGURE 9.7: Satellite test problem reconstructions: (a) PSF computed with CSeB-A (SNR=16.246), (b) Image computed with CSeB-A (SNR=11.856), (c) Image computed with Tikhonov-TV (SNR=10.816).

white Gaussian noise to the PSF, so that $\epsilon = 0.7\,\|k\|$. All the corresponding images are shown in Figure 9.6.

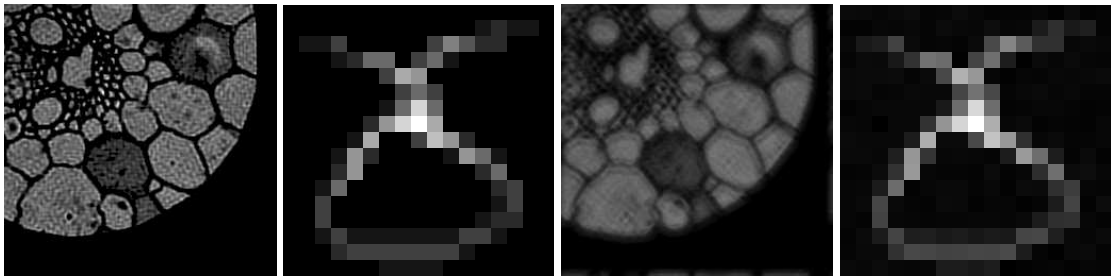
We compare CSeB-A with Tikhonov-TV, where for both methods we fix $\alpha = 10^{-6}$ and for CSeB-A we set $\beta = 10^{-4}$. In Figure 9.7 we can see the reconstructed images for both methods and the corresponding SNR. It can be noted that our proposal allows a large improvements in the details of the reconstruction.

In Figure 9.10 the norms of the iterates against the given bounds are shown. As in the previous example the norms are much smaller than the bounds and stabilize quickly.

Grain We now test our method on the Grain image blurred with a non symmetric PSF. We add white Gaussian noise such that $\xi = 0.01$ and we also add white Gaussian noise to the PSF, so that $\epsilon = 0.8\,\|k\|$. All the corresponding images are shown in Figure 9.8.

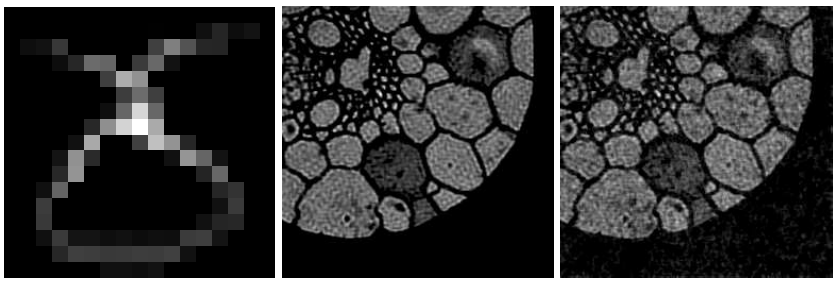
Similarly to the previous example the reconstruction obtained with CSeB-A is more accurate than the one obtained with Tikhonov-TV, see the SNR and the restored images in Figure 9.9, where we compare again CSeB-A with Tikhonov-TV fixing $\alpha = 10^{-4}$ for both methods and $\beta = 10^{-4}$ for CSeB-A.

Finally in Figure 9.10 we can see the norms of the iterates compared with the given bounds. Also in this case the bounds are respected and the norms of the iterates stabilize in few iterations.



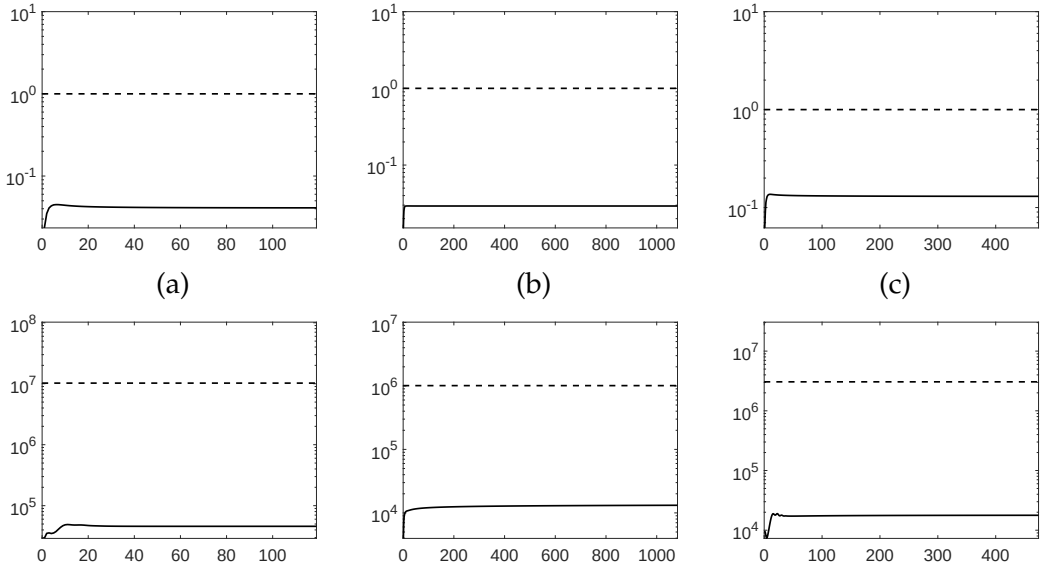
(a) (b) (c) (d)

FIGURE 9.8: Grain test problem: (a) Test image, (b) Noise-free PSF, (c) Blurred and noisy image, (d) Noisy PSF.



(a) (b) (c)

FIGURE 9.9: Grain test problem reconstructions: (a) PSF computed with SeB-A (SNR = 22.257), (b) Image computed with SeB-A (SNR = 20.731), (c) Image computed with Tikhonov-TV (SNR = 12.103).



(a) (b) (c)

(d) (e) (f)

FIGURE 9.10: Comparison of the norm of the iterates generated by CSeB-A with the proposed bounds for all the examples. The first row are related to the PSFs, the second to the images. The first column are the norms for the first example with 8% of noise, the second column is related to the Satellite example and the last one is for the Grain example. The solid curve is the Euclidean norm of the iterates, the dashed curve is the bound.

Chapter 10

Conclusions and Future work

We now draw some conclusions on the presented work and describe some possible future extensions.

In this thesis we dealt with ill-posed inverse problems and described several algorithms to compute in a fast way accurate approximations of the solution of these problems.

All the presented work stemmed from the basic idea of Tikhonov regularization both in its iterated and non-iterated form. We have shown that, by inserting inside the algorithms available knowledge on the solution, we are able to greatly improve the quality of the reconstructions while keeping the computational cost under control.

In Chapter 3 we saw that simply introducing the nonnegativity constraint inside the standard Tikhonov regularization can improve the quality of the reconstructions. Moreover, taking advantage of the very rapid decay of the singular values, it is possible to use the Krylov space theory to achieve very fast computations without losing anything in term of quality of the reconstruction.

In Chapter 4 we saw that the introduction of a regularization operator different from the identity inside the iterated Tikhonov algorithm can enhance the quality of the reconstructions. If we are able to select a regularization operator such that significant components of the exact solution lie in its null space then this can dramatically improve the performances of the algorithm.

Combining this two observation in Chapter 6 we were able to extend the AIT method, which was already a very powerful algorithm. By inserting either the general form of Tikhonov regularization or the projection inside a closed and convex set we created two highly performing algorithms. We were able to prove their theoretical properties, like the monotonic decay of the reconstruction error and their regularization effect. Moreover, we formulated an algorithm which combines both the presence of the regularization operator and the projection into a closed and convex set. Even if we were not able to provide a precise proof of convergence we gave numerical evidences of the potentialities of this algorithm.

In Chapter 5, taking two different extensions of the classical Tikhonov regularization method, we formulated two iterative regularization methods. We gave a very deep theoretical insight on their convergence properties, including optimality results. We were able to derive also the nonstationary version of these algorithms and we gave conditions on the parameters for the convergence.

In all the above mentioned methods, either by using the knowledge of the noise level or by exploiting the nonstationarity we were able to build methods which do not require the manual estimate of any constant, but derive the necessary parameters in an automatic way. This does not only means that the methods formulated are stable and robust, but also that

can be considered for real data and not only for synthetic experiments. Even though we did not present any real case scenario for these methods the absence of any parameter estimation makes their usage easy on real data.

All the methods proposed in Chapters 3-6 only relied on purely linear techniques. From Chapter 7 we started to insert non-linear elements in our algorithms.

In Chapter 7 we combined a non-linear framelet denoising with a multigrid algorithm designed for image deblurring. In particular we used one of the algorithms described in Chapter 6 as post-smoother and soft-thresholding denoising algorithm as pre-smoother. This choice made us able to construct a multigrid algorithm with regularizing properties, which we were able to prove, under reasonable hypothesis. We constructed an algorithm which is able to recover with high accuracy images from blurred measurements. The presence of the non-linear denoising let us recover the edges of the image without reconstructing the noise. Moreover, since we employed a nonstationary strategy for both the pre- and post-smoother, the formulated algorithm is very robust and does not need the estimation of any parameter.-

In Chapter 8 we developed a method tailored for an application in optics starting from the non-linear Lucy-Richardson algorithm. We added a weak constraint inside the algorithm. By weak we mean that the computed solution does not necessarily satisfy this constraint but the divergence from this is used as a penalization term, like in Tikhonov. The weight of this penalty term is determined by a parameter that can be estimated using a second constraint known on the true solution. We applied this "hybrid" approach since in real case scenarios the value of the first constraint can be evaluated precisely, whereas the measurement of the second one is likely to be affected by heavy noise. But, thanks to the robustness of the LR algorithm, the noise on the second constraint does not affect too much the quality of the reconstruction.

Finally, in Chapter 9, we introduced a slightly more complicated inverse problem. In this case not only the solution of the problem is unknown, but also the operator itself has a certain degree of uncertainty on it. In particular we assumed that the operator depends on a set of parameters which is not completely known, more precisely, the parameters given by the problem are affected by noise. We formulated a functional that depends on both the parameters and the solution of the problem and looked for its minimum. Being the functional non-convex, the existence of a global minimum was not assured. Nevertheless, we proved the existence of a global minimum and we were able to provide stability and regularization properties. The computation of a stationary point of a non-convex functional is no easy task. We used the well known ADMM algorithm to solve this problem. However, since the convergence of such an algorithm is not assured in the non-convex case, we provided a proof of convergence under some assumptions and a conjecture. We did not provide any proof for the conjecture, however in the numerical it is always largely satisfied.

With the work done to this point we were able to construct accurate, robust, and fast algorithm for ill-posed inverse problems. However, there is still room for improvements and future studies for all the proposed methods.

As we saw in Chapter 3 the nonnegative constraint can help in providing accurate solutions. Combining active set methods with the modulus method could, at least in principle, helps in achieving even better solutions. Both the Modulus Method and the active set methods have their shortcomings. In particular we saw that MM has some issues when dealing with solutions with huge zero areas, on the other hand active set method usually suffers of slow convergence when the noise level is low. By combining the two approaches it may be possible to cancel both of this issues and create a more robust method.

In Chapter 4 we formulated a nonstattionary algorithm. By nonstattionary in this setting we meant that the regularization parameter can change throughout the iterations. However, it is possible to think about a "different" nonstattionary. In particular we would like to see what happens when not only the regularization parameter changes at each iteration but also the regularization operator itself can vary. In particular, one might think of inserting more and more information on the solution in the regularization operator as the iterations goes on.

Another possible extension of the work we have done would be to combine the methods in Chapters 5 and 6. Combining the various strategy used in the algorithms of Chapter 6 and the fractional or weighted filters of Chapter 5 we could develop an algorithm that is both fast and stable. Moreover, such an algorithm should be able to well recover edges without amplifying the noise.

The work in Chapter 7 can be seen as a particular version of the iterative soft thresholding algorithm (ISTA) in [43] where the Landweber step has been substituted with a step of multigrid. The evolution of the ISTA algorithm is the Linearized Bregman Splitting method. Then it is only natural to think about a Linearized Bregman Splitting where the Landweber step is substituted with a step of our multigrid algorithm. A similar approach has already been used in [34] where the Landweber step has been substituted with a step of the AIT algorithm.

The method developed in Chapter 8 was designed to work with only one source of data on the particles we are interested in studying. However, new technological advances let us measure different quantities about the same sample simultaneously. Combining the two different sources of data inside a LR framework could let us have a more stable, robust and accurate algorithm that is also able to exploit additional information experimentally measurable on the sample.

Finally, since the algorithm proposed in Chapter 9 does not have any rule for estimating the regularization parameters, it would be interesting to develop some criterion to determine a good choice of parameters, maybe exploiting the knowledge of the levels of noise δ and ϵ.

Summarizing we think that we have shown a collection of nice methods that all stem from the basic idea of Tikhonov regularization. Moreover, all these can lead to interesting future development and open new questions that need to be answered.

Bibliography

[1] Mariana SC Almeida and Luís B Almeida. "Blind and Semi-Blind Deblurring of Natural Images". In: *IEEE Transactions on Image Processing* 19.1 (2010), pp. 36–52. ISSN: 1057-7149.

[2] Mariana SC Almeida and Mário AT Figueiredo. "Deconvolving images with unknown boundaries using the alternating direction method of multipliers". In: *IEEE Transactions on Image Processing* 22.8 (2013), pp. 3074–3086.

[3] Mariana SC Almeida and Mário AT Figueiredo. "Frame-based image deblurring with unknown boundary conditions using the alternating direction method of multipliers". In: *2013 IEEE International Conference on Image Processing*. IEEE. 2013, pp. 582–585.

[4] Antonio Aricò and Marco Donatelli. "A V-cycle Multigrid for multilevel matrix algebras: proof of optimality". In: *Numerische Mathematik* 105.4 (2007), pp. 511–547.

[5] Antonio Aricò, Marco Donatelli, James G Nagy, and Stefano Serra-Capizzano. "The anti-reflective transform and regularization by filtering". In: *Numerical Linear Algebra in Signals, Systems and Control*. Springer, 2011, pp. 1–21.

[6] James Baglama and Lothar Reichel. "Augmented implicitly restarted Lanczos bidiagonalization methods". In: *SIAM Journal on Scientific Computing* 27.1 (2005), pp. 19–42.

[7] Zhong-Zhi Bai. "Modulus-based matrix splitting iteration methods for linear complementarity problems". In: *Numerical Linear Algebra with Applications* 17.6 (2010), pp. 917–933.

[8] Zhong-Zhi Bai, Alessandro Buccini, Ken Hayami, Lothar Reichel, Jun-Feng Yin, and Ning Zheng. "Modulus-Based Method for Constrained Tikhonov Regularization". In: *Under Review on: Journal of Computational Mathematics* (2016).

[9] Zhong-Zhi Bai, Gene H Golub, and Michael K Ng. "Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems". In: *SIAM Journal on Matrix Analysis and Applications* 24.3 (2003), pp. 603–626.

[10] Zhong-Zhi Bai and Li-Li Zhang. "Modulus-based synchronous multisplitting iteration methods for linear complementarity problems". In: *Numerical Linear Algebra with Applications* 20.3 (2013), pp. 425–439.

[11] Amir Beck, Aharon Ben-Tal, and Christian Kanzow. "A fast method for finding the global solution of the regularized structured total least squares problem for image deblurring". In: *SIAM J. Matrix Anal. Appl.* 30.1 (2008), pp. 419–443. ISSN: 0895-4798.

[12] Sebastian Berisha and James G Nagy. *Iterative Methods for Image Restoration*. Tech. rep. http://www.mathcs.emory.edu/~nagy/RestoreTools/IR.pdf. Department of Mahtematics and Computer Science, Emory University, 2012.

[13] Mario Bertero and Patrizia Boccacci. *Introduction to inverse problems in imaging*. CRC press, 1998.

[14] Davide Bianchi, Alessandro Buccini, Marco Donatelli, and Stefano Serra-Capizzano. "Iterated fractional Tikhonov regularization". In: *Inverse Problems* 31.5 (2015), p. 055005.

[15] José M Bioucas-Dias and Mário AT Figueiredo. "A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration". In: *Image Processing, IEEE Transactions on* 16.12 (2007), pp. 2992–3004.

[16] Åke Björck. *Numerical methods for least squares problems*. Siam, 1996.
[17] Ismael Rodrigo Bleyer and Ronny Ramlau. "A double regularization approach for inverse problems with noisy data and inexact operator". In: *Inverse Problems* 29.2 (2013), p. 025004.
[18] Ismael Rodrigo Bleyer and Ronny Ramlau. "An alternating iterative minimization algorithm for the double-regularised total least square functional". In: *Inverse Problems* 31.7 (2015), p. 075004.
[19] Craig F Bohren and Dan E Hirleman. "Feature on Optical Particle Sizing". In: *Appl. Opt.* 30 (1991), pp. 4685–4987.
[20] Silvia Bonettini, Anastasia Cornelio, and Marco Prato. "A new semiblind deconvolution approach for fourier-based image restoration: An application in astronomy". In: *SIAM Journal on Imaging Sciences* 6.3 (2013), pp. 1736–1757.
[21] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers". In: *Found. Trends Mach. Learn.* 3.1 (Jan. 2011), pp. 1–122. ISSN: 1935-8237.
[22] Achi Brandt. "Multi-Level Adaptive Solutions to Boundary-Value Problems". In: *Mathematics of Computation* 31.138 (1977), pp. 333–390.
[23] Haim Brezis. *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media, 2010.
[24] William L Briggs, Van Emden Henson, and Stephen F McCormick. *A Multigrid Tutorial, Second Edition*. Second. Society for Industrial and Applied Mathematics, 2000.
[25] Markus Brill and Eberhard Schock. "Iterative solution of ill-posed problems-a survey". In: *Model Optimization in Exploration Geophysics* 1 (1987), pp. 17–37.
[26] Alessandro Buccini. "Regularizing preconditioners by non-stationary iterated Tikhonov with general penalty term". In: *Applied Numerical Mathematics* (2016).
[27] Alessandro Buccini and Marco Donatelli. "Multigrid iterative regularization method for image deblurring with arbitrary boundary conditions". In: *In Progress* (2016).
[28] Alessandro Buccini, Marco Donatelli, and Fabio Ferri. "Weakly constrained Lucy-Richardson with application to inverse light scattering". In: *Submitted* (2016).
[29] Alessandro Buccini, Marco Donatelli, and Ronny Ramlau. "A semi-blind regularization algorithm for inverse problems with application to image deblurring". In: *Submitted* (2016).
[30] Alessandro Buccini, Marco Donatelli, and Lothar Reichel. "Iterated Tikhonov with general penalty term". In: *Accepted on: Numerical Linear Algebra and Applications* (2016).
[31] Jian-Feng Cai, Raymond H Chan, and Zuowei Shen. "A framelet-based image inpainting algorithm". In: *Applied and Computational Harmonic Analysis* 24.2 (2008), pp. 131–149.
[32] Jian-Feng Cai, Stanley Osher, and Zuowei Shen. "Linearized Bregman iterations for frame-based image deblurring". In: *SIAM Journal on Imaging Sciences* 2.1 (2009), pp. 226–252.
[33] Jian-Feng Cai, Stanley Osher, and Zuowei Shen. "Split Bregman methods and frame based image restoration". In: *Multiscale modeling & simulation* 8.2 (2009), pp. 337–369.
[34] Yuantao Cai, Marco Donatelli, Davide Bianchi, and Ting-Zhu Huang. "Regularization preconditioners for frame-based image deblurring with reduced boundary artifacts". In: *SIAM Journal on Scientific Computing* 38.1 (2016), B164–B189.
[35] Daniela Calvetti, Bryan W Lewis, Lothar Reichel, and Fiorella Sgallari. "Tikhonov regularization with nonnegativity constraint". In: *Electronic Transactions on Numerical Analysis* 18 (2004), pp. 153–173.

[36] Raymond H Chan, Tony F Chan, and W. L Wan. "Multigrid for Differential-Convolution Problems Arising from Image Processing". In: *Processing, in Proceedings of the Workshop on Sci. Comput*. Springer-Verlag, 1997, pp. 58–72.

[37] Raymond H Chan and Ke Chen. "A Multilevel Algorithm for Simultaneously Denoising and Deblurring Images". In: *SIAM Journal on Scientific Computing* 32.2 (2010), pp. 1043–1063.

[38] Raymond H Chan, Min Tao, and Xiaoming Yuan. "Constrained total variation deblurring models and fast algorithms based on alternating direction method of multipliers". In: *SIAM Journal on imaging Sciences* 6.1 (2013), pp. 680–697.

[39] Tony F Chan and Chiu-Kwong Wong. "Total variation blind deconvolution". In: *IEEE Transactions on Image Processing* 7.3 (1998), pp. 370–375. ISSN: 1057-7149.

[40] Julianne Chung, James G Nagy, and Dianne P O'Leary. "A weighted GCV method for Lanczos hybrid regularization". In: *Electronic Transactions on Numerical Analysis* 28 (2008), pp. 149–167.

[41] Anastasia Cornelio, Federica Porta, and Marco Prato. "A convergent least-squares regularized blind deconvolution approach". In: *Applied Mathematics and Computation* 259 (2015), pp. 173–186.

[42] Richard W Cottle and Jongshi Pang. *RE Stone.(1992). The linear complementarity problem*.

[43] Ingrid Daubechies, Michel Defrise, and Christine De Mol. "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint". In: *Communications on pure and applied mathematics* 57.11 (2004), pp. 1413–1457.

[44] Pietro Dell'Acqua, Marco Donatelli, and Claudio Estatico. "Preconditioners for image restoration by reblurring techniques". In: *Journal of Computational and Applied Mathematics* 272 (2014), pp. 313–333.

[45] Marco Donatelli. "An iterative multigrid regularization method for Toeplitz discrete ill-posed problems". In: *Numerical Mathematics: Theory, Methods and Applications* 5.01 (2012), pp. 43–61.

[46] Marco Donatelli. "On nondecreasing sequences of regularization parameters for nonstationary iterated Tikhonov". In: *Numerical Algorithms* 60.4 (2012), pp. 651–668.

[47] Marco Donatelli, Claudio Estatico, Andrea Martinelli, and Stefano Serra-Capizzano. "Improved image deblurring with anti-reflective boundary conditions and re-blurring". In: *Inverse problems* 22.6 (2006), p. 2035.

[48] Marco Donatelli, Claudio Estatico, James G Nagy, Lisa Perrone, and Stefano Serra-Capizzano. "Anti-reflective boundary conditions and fast 2D deblurring models". In: *Optical Science and Technology, SPIE's 48th Annual Meeting*. International Society for Optics and Photonics. 2003, pp. 380–389.

[49] Marco Donatelli and Martin Hanke. "Fast nonstationary preconditioned iterative methods for ill-posed problems, with application to image deblurring". In: *Inverse Problems* 29.9 (2013), p. 095008.

[50] Marco Donatelli, David Martin, and Lothar Reichel. "Arnoldi methods for image deblurring with anti-reflective boundary conditions". In: *Applied Mathematics and Computation* 253 (2015), pp. 135–150.

[51] Marco Donatelli and Nicola Mastronardi. "Fast deconvolution with approximated PSF by RSTLS with antireflective boundary conditions". In: *J. Comput. Appl. Math.* 236.16 (2012), pp. 3992–4005. ISSN: 0377-0427.

[52] Marco Donatelli, Arthur Neuman, and Lothar Reichel. "Square regularization matrices for large linear discrete ill-posed problems". In: *Numerical Linear Algebra with Applications* 19.6 (2012), pp. 896–913.

[53] Marco Donatelli and Lothar Reichel. "Square smoothing regularization matrices with accurate boundary conditions". In: *Journal of Computational and Applied Mathematics* 272 (2014), pp. 334–349.

[54] Marco Donatelli and Stefano Serra-Capizzano. "Antireflective boundary conditions for deblurring problems". In: *Journal of Electrical and Computer Engineering* 2010 (2010), p. 2.

[55] Marco Donatelli and Stefano Serra-Capizzano. "Filter factor analysis of an iterative multilevel regularizing method". In: *Electron. Trans. Numer. Anal.* 29 (2007/2008), pp. 163–177.

[56] Marco Donatelli and Stefano Serra-Capizzano. "On the regularizing power of multigrid-type algorithms". In: *SIAM Journal on Scientific Computing* 27.6 (2006), pp. 2053–2076.

[57] Jun-Liang Dong and Mei-Qun Jiang. "A modified modulus method for symmetric positive-definite linear complementarity problems". In: *Numerical Linear Algebra with Applications* 16.2 (2009), pp. 129–143.

[58] David L Donoho. "De-noising by soft-thresholding". In: *IEEE transactions on information theory* 41.3 (1995), pp. 613–627.

[59] Lars Eldén. "A weighted pseudoinverse, generalized singular values, and constrained least squares problems". In: *BIT Numerical Mathematics* 22.4 (1982), pp. 487–502.

[60] Lars Eldén. "Algorithms for the regularization of ill-conditioned least squares problems". In: *BIT Numerical Mathematics* 17.2 (1977), pp. 134–145.

[61] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*. Vol. 375. Springer Science & Business Media, 1996.

[62] Malena I Espanol and Misha E Kilmer. "A Wavelet-Based Multilevel Approach for Blind Deconvolution Problems". In: *SIAM Journal on Scientific Computing* 36.4 (2014), A1432–A1450.

[63] Malena I Espanol and Misha E Kilmer. "Multilevel approach for signal restoration problems with Toeplitz matrices". In: *SIAM Journal on Scientific Computing* 32.1 (2010), pp. 299–319.

[64] Caterina Fenu, Lothar Reichel, and Giuseppe Rodriguez. "GCV for Tikhonov regularization via global Golub–Kahan decomposition". In: *Numerical Linear Algebra with Applications* 23.3 (2016), pp. 467–484.

[65] Fabio Ferri, Gabriella Righini, and Enrico Paganini. "Inversion of low-angle elastic light-scattering data with a new method devised by modification of the Chahine algorithm". In: *Appl. Opt.* 36.30 (1997), pp. 7539–7550.

[66] David A Fish, John G Walker, A. M Brinicombe, and Edward R Pike. "Blind deconvolution by means of the Richardson–Lucy algorithm". In: *J. Opt. Soc. Am. A* 12.1 (1995), pp. 58–65.

[67] Silvia Gazzola and James G Nagy. "Generalized Arnoldi–Tikhonov Method for Sparse Reconstruction". In: *SIAM Journal on Scientific Computing* 36.2 (2014), B225–B247.

[68] Silvia Gazzola, Paolo Novati, and Maria Rosaria Russo. "On Krylov projection methods and Tikhonov regularization". In: *Electronic Transactions on Numerical Analysis* 44 (2015), pp. 83–123.

[69] Daniel Gerth, Esther Klann, Ronny Ramlau, and Lothar Reichel. "On fractional Tikhonov regularization". In: *Journal of Inverse and Ill-Posed Problems* 23.6 (2015), pp. 611–625.

[70] Otto Glatter. "Fourier transformation and deconvolution". In: *Neutrons, X-rays and light: Scattering methods applied to soft condensed matter*. Ed. by Patrik Linder and Thomas Zemb. North Holland, Elsevier, 2002.

[71] Otto Glatter. "Static light scattering of large systems". In: *Neutrons, X-rays and light: Scattering methods applied to soft condensed matter*. Ed. by Patrik Linder and Thomas Zemb. North Holland, Elsevier, 2002.

[72] Otto Glatter. "The inverse scattering problem in small angle scattering". In: *Neutrons, X-rays and light: Scattering methods applied to soft condensed matter*. Ed. by Patrik Linder and Thomas Zemb. North Holland, Elsevier, 2002.

[73] Gene H Golub and Charles F Van Loan. *Matrix Computations, 4th. edition*. Vol. 3. JHU Press, 2013.

[74] Gérard Gouesbet and Gérard Gréhan. *Optical particle sizing: theory and practice*. Academic press, New York, 1969.

[75] Charles W Groetsch. *The theory of Tikhonov regularization for Fredholm equations of the first kind*. Vol. 105. Pitman Advanced Publishing Program, 1984.

[76] Apostolos Hadjidimos and Michael G Tzoumas. "Nonstationary extrapolated modulus algorithms for the solution of the linear complementarity problem". In: *Linear Algebra and its Applications* 431.1 (2009), pp. 197–210.

[77] Martin Hanke. "A regularizing Levenberg-Marquardt scheme, with applications to inverse groundwater filtration problems". In: *Inverse problems* 13.1 (1997), p. 79.

[78] Martin Hanke. *Conjugate gradient type methods for ill-posed problems*. Vol. 327. CRC Press, 1995.

[79] Martin Hanke and Charles W Groetsch. "Nonstationary iterated Tikhonov regularization". In: *Journal of Optimization Theory and Applications* 98.1 (1998), pp. 37–53.

[80] Martin Hanke and Per Christian Hansen. "Regularization methods for large-scale problems". In: *Surv. Math. Ind* 3.4 (1993), pp. 253–315.

[81] Martin Hanke and R. Curtis Vogel. "Two-level preconditioners for regularized inverse problems I: Theory". In: *Numerische Mathematik* 83.3 (1999), pp. 385–402.

[82] Per Christian Hansen. *Rank Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. SIAM, 1998.

[83] Per Christian Hansen. "Regularization tools version 4.0 for Matlab 7.3". In: *Numerical algorithms* 46.2 (2007), pp. 189–194.

[84] Per Christian Hansen, James G Nagy, and Dianne P O'leary. *Deblurring images: matrices, spectra, and filtering*. Vol. 3. Siam, 2006.

[85] Lin He, Antonio Marquina, and Stanley J. Osher. "Blind deconvolution using TV regularization and Bregman iteration". In: *International Journal of Imaging Systems and Technology* 15.1 (2005), pp. 74–83. ISSN: 1098-1098.

[86] Michiel E Hochstenbach and Lothar Reichel. "Fractional Tikhonov regularization for linear discrete ill-posed problems". In: *BIT Numerical Mathematics* 51.1 (2011), pp. 197–215.

[87] Mingyi Hong, Zhi-Quan Luo, and Meisam Razaviyayn. "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems". In: *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE. 2015, pp. 3836–3840.

[88] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[89] Guangxin Huang, Lothar Reichel, and Feng Yin. "On the choice of solution subspace for nonstationary iterated Tikhonov regularization". In: *Numerical Algorithms* (In press).

[90] Guangxin Huang, Lothar Reichel, and Feng Yin. "Projected nonstationary iterated Tikhonov regularization". In: *BIT Numerical Mathematics* 56.2 (2016), pp. 467–487.

[91] Hendrik C Van de Hulst. *Light scattering by small particles*. New York: Dover Publications, 1981. Chap. 9, p. 127.

[92] Barbara Kaltenbacher. "On the regularizing properties of a full multigrid method for ill-posed problems". In: *Inverse Problems* 17.4 (2001), p. 767.

[93] Milton Kerker. *The scattering of light and other electromagnetic radiation*. Academic press, New York, 1969.

[94] Thomas J King. "Multilevel algorithms for ill-posed problems". In: *Numerische Mathematik* 61.1 (1992), pp. 311–334.
[95] Esther Klann and Ronny Ramlau. "Regularization by fractional filter methods and data smoothing". In: *Inverse Problems* 24.2 (2008), p. 025018.
[96] Alfred K Louis. *Inverse und schlecht gestellte Probleme*. Teubner, Stuttgart, 1989.
[97] Leon B Lucy. "An iterative technique for the rectification of observed distributions". In: *AJ* 79 (1974), p. 745.
[98] Serena Morigi, Robert J Plemmons, Lothar Reichel, and Fiorella Sgallari. "A hybrid multilevel-active set method for large box-constrained linear discrete ill-posed problems". In: *Calcolo* 48.1 (2011), pp. 89–105.
[99] Serena Morigi, Lothar Reichel, and Fiorella Sgallari. "An interior-point method for large constrained discrete ill-posed problems". In: *Journal of computational and applied mathematics* 233.5 (2010), pp. 1288–1297.
[100] Serena Morigi, Lothar Reichel, Fiorella Sgallari, and Andriy Shyshkov. "Cascadic multiresolution methods for image deblurring". In: *SIAM Journal on Imaging Sciences* 1.1 (2008), pp. 51–74.
[101] Serena Morigi, Lothar Reichel, Fiorella Sgallari, and Fabiana Zama. "An iterative method for linear discrete ill-posed problems with box constraints". In: *Journal of Computational and Applied Mathematics* 198.2 (2007), pp. 505–520.
[102] James G Nagy and Zdenek Strakos. "Enforcing nonnegativity in image reconstruction algorithms". In: *International Symposium on Optical Science and Technology*. International Society for Optics and Photonics. 2000, pp. 182–190.
[103] Artem Napov and Yvan Notay. "When does two-grid optimality carry over to the V-cycle?" In: *Numerical Linear Algebra with Applications* 17.2-3 (2010), pp. 273–290.
[104] Arthur Neuman, Lothar Reichel, and Hassane Sadok. "Algorithms for range restricted iterative methods for linear discrete ill-posed problems". In: *Numerical Algorithms* 59.2 (2012), pp. 325–331.
[105] Michael K Ng, Raymond H Chan, and Wun-Cheung Tang. "A fast algorithm for deblurring models with Neumann boundary conditions". In: *SIAM Journal on Scientific Computing* 21.3 (1999), pp. 851–866.
[106] Jorge Nocedal and Stephen J Wright. "Numerical optimization 2nd". In: (2006).
[107] E Onunwor and Lothar Reichel. "Modulus-Type Inner Outer Iterative Methods for Nonnegative Constrained Least Squares Problems". In: *Submitted* (2016).
[108] Roger Penrose. "A generalized inverse for matrices". In: *Mathematical proceedings of the Cambridge philosophical society*. Vol. 51. 03. Cambridge Univ Press. 1955, pp. 406–413.
[109] Marco Prato, Andrea La Camera, Silvia Bonettini, Simone Rebegoldi, Mario Bertero, and Patrizia Boccacci. "A blind deconvolution method for ground based telescopes and Fizeau interferometers". In: *New Astronomy* 40 (2015), pp. 1–13.
[110] Lothar Reichel and Andriy Shyshkov. "Cascadic multilevel methods for ill-posed problems". In: *Journal of Computational and Applied Mathematics* 233.5 (2010), pp. 1314 –1325.
[111] Lothar Reichel and Qiang Ye. "Simple square smoothing regularization operators". In: *Electronic Transactions on Numerical Analysis* 33 (2009), pp. 63–83.
[112] Lothar Reichel and Xuebo Yu. "Matrix decompositions for Tikhonov regularization". In: *Electronic Transactions on Numerical Analysis* 43 (2015), pp. 223–243.
[113] William Hadley Richardson. "Bayesian-Based Iterative Method of Image Restoration". In: *JOSA* 62.1 (1972), pp. 55–59.
[114] Andreas Rieder. "A wavelet multilevel method for ill-posed problems stabilized by Tikhonov regularization". In: *Numerische Mathematik* 75.4 (1997), pp. 501–522.

[115] Marielba Rojas and Trond Steihaug. "An interior-point trust-region-based method for large-scale non-negative regularization". In: *Inverse Problems* 18.5 (2002), p. 1291.

[116] Walter Rudin. *Functional analysis. International series in pure and applied mathematics*. McGraw-Hill, Inc., New York, 1991.

[117] Walter Rudin. *Real and complex analysis*. Tata McGraw-Hill Education, 1987.

[118] John W Ruge and Klaus Stuben. "Algebraic Multigrid". In: *Multigrid Methods*. Ed. by Stephen F Mccormick. Philidelphia, Pennsylvania: SIAM, 1987. Chap. 4, pp. 73–130.

[119] Stefano Serra-Capizzano. "A note on antireflective boundary conditions and fast deblurring models". In: *SIAM Journal on Scientific Computing* 25.4 (2004), pp. 1307–1325.

[120] Ulrich Trottenberg, Cornelius W Oosterlee, and Anton Schuller. *Multigrid*. Academic press, 2000.

[121] Eduardo H Zarantonello. *Projections on convex sets in Hilbert space and spectral theory*. University of Wisconsin, 1971.

[122] Ning Zheng, Ken Hayami, and Jun-Feng Yin. "Modulus-Type Inner Outer Iteration Methods for Nonnegative Constrained Least Squares Problems". In: *SIAM Journal on Matrix Analysis and Applications* 37 (2016), pp. 1250–1278.